Camera-based distance estimation for autonomous vehicles

Master Thesis
Technical University Berlin
Faculty Electrical Engineering and Computer Science
Department Computer Vision and Remote Sensing
Field of Study Computer Engineering

Submitted on:

Author: Patrick Irmisch <patrickirmisch@web.de>
MatrNr. 339280

1. Evaluator: Prof. Dr.-Ing. Olaf Hellwich
2. Evaluator: Prof. Dr. Oliver Brock

Thesis supervised by: Dr. rer. nat. Jürgen Wohlfeil
and Dr. Eugen Funk
German Aerospace Center (DLR)
Institute of Optical Sensor Systems

Eidesstattliche Erklärung

Die selbständige und eigenhändige Anfertigung versichert an Eides statt
Berlin, den

.....
Unterschrift

Abstract

The aim of this work is the investigation of camera-based techniques for distance estimation between two autonomous vehicles. While both monocular- and stereo-camera methods are explored, this study focuses on the usage of fiducial markers.
Therefore, existing fiducial markers are discussed and selected. Based on this selection, three configurations of markers are proposed and applied to different distance estimation methods. The chosen markers are AprilTag and WhyCon. Their distances are estimated by means of Perspective-n-Point, 3D position calculation of a circle and stereo-based triangulation.
Within this study the presented methods are evaluated based on their distance estimation accuracy and applicable range. They are compared with each other and with the common stereo method Semi-Global-Matching. Moreover, the influence of uncertainties is explored with reference to geometrical calibration. A setup is presented to evaluate the techniques based on real-world and simulated data. In order to gain insights on the methods properties, a simulation is used that facilitates variation of the image data. In addition, a Monte-Carlo-Simulation allows to model calibration uncertainty. The obtained observations are substantiated based on two real-world experiments.
The results demonstrate the potential of fiducial markers for relative distance estimation of vehicles in terms of high accuracy and low uncertainty. The lower sensitivity to uncertainties in camera calibration makes fiducial markers preferable to stereo methods.

Zusammenfassung

Die Masterarbeit untersucht Verfahren zur relativen Distanzbestimmung zwischen zwei autonomen Fahrzeugen mit Hilfe von Monokular- und Stereokameras. Hierbei liegt der Fokus der Studie auf der Anwendung von Markierungen bekannter Größe.
Dafür werden verschiedene existierende Bezugsmarker (eng.: fiducial marker) in Betracht gezogen und in drei verschiedenen Konfigurationen mit unterschiedlichen Verfahren zur Distanzbestimmung angewandt. Die verwendeten Marker sind AprilTag und WhyCon. Ihre Distanzen werden berechnet mit Hilfe der Methoden: Lösung des Perspective-n-Point Problems, 3D-Positionsberechung eines Kreises und Triangulation in einem Stereokamerasystem.
Die ausgewählten Methoden werden anhand der Genauigkeit ihrer Abstandsberechnung und ihrer möglichen Anwendungsreichweite miteinander verglichen. Sie werden zusätzlich gegen das anerkannte Stereoverfahren Semi-Global-Matching gehalten. Des Weiteren werden die resultierenden Ungenauigkeiten untersucht, die durch Unsicherheiten in geometrischen Kalibrierungsparametern entstehen. Hierfür wird ein Evaluierungskonzept vorgeschlagen und umgesetzt, welches sowohl reale Daten als auch simulierte Daten verwendet. Um Einblicke in das Verhalten der Methoden zu erlangen wird eine Simulation verwendet, die eine Variation in den Bilddaten ermöglicht. Eine Monte-Carlo-Simulation ermöglicht eine Modellierung von Kalibrierungsunsicherheiten. Um die Beobachtungen zu untermauern, werden zwei in der realen Welt durchgeführte Experimente verwendet.
Die Ergebnisse zeigen das Potential von Marker-basierten Methoden zur Bestimmung von relativen Distanzen. Im direkten Vergleich zu Stereoverfahren sind sie aufgrund ihrer geringeren Sensibilität gegenüber den Unsicherheiten von Kamerakalibrierungsparametern zu bevorzugen.

Contents

List of Figures

List of Tables

Register of listings

Nomenclature

CAD ..... Computer-Aided Design

DLR ..... German Aerospace Center

fpb ..... frames per box

GPS ..... Global Positioning System

LED ..... Light Emitting Diode

LIDAR ..... Light Detection and Ranging System

MCS ..... Monte-Carlo-Simulation

NGT ..... Next Generation Train

PDF ..... Probability Density Function

Pixel, px ..... Picture Element

PRNU ..... Photo Response Non Uniformity

RPV ..... Relative Positioning of Vehicles

SGM ..... Semi-Global-Matching

1 Introduction

This chapter provides a motivation for this work and a description of the objective, formulated by research questions. Then the general structure of the work is delineated.

1.1 Motivation

Relative distance estimation incarnates an important role in numerous safety-critical applications in the context of Advanced Driver Assistance Systems. Thus, systems as Automatic Cruise Control in vehicle-platooning applications rely on continuous knowledge about the relative position of the preceding vehicle. In this context, a future-oriented application is investigated by the German Aerospace Center with "Virtual Coupling" of trains in the project Next Generation Train (DLR, 2016, NGT). The vision is to replace physical coupling by driving in short distances as illustrated in Figure 1. This allows to compose multiple trains during continuous driving, which promises a more effective use of the rail system and shortened travel times.
Certainly, the estimation of the relative distance to the preceding train needs to be highly reliable and accurate. To accomplish this task while ensuring flexibility and independence to external systems, such as Global Positioning System (GPS), vehicle-based sensors are commonly used. Most represented are radar sensors, Light Detection and Ranging (LIDAR) devices as well as camera systems. These sensors show different advantages and disadvantages, which is why safety-critical systems are usually designed to combine results of different sensors. However, in recent years camera systems enjoy increasing attention due to more available computing power and rapid advances in camera technology. Moreover, camera systems come along with different methods to estimate the relative distance. For instance, they are differentiated in monocular- or stereo systems and based on image features, natural features or fiducial features. Due to this versatility, this thesis investigates the use of camera sensors.
Before applying methods to a safety-critical real-world scenario they have to be examined on their reliability, accuracy and sensitivity to various influences. In general, real-world experiments represent the key for a conclusive investigation. Nonetheless, real-world experiments do not allow to analyze the dependency on complex variations of the input parameters such as calibration uncertainty. Because of that, this work presents an evaluation setup that evaluates real-world data and supports these experiments by simulated data based on a Monte-Carlo-Simulation.

Figure 1: Simulation based illustration of "Virtual Coupling" of trains [Blend1]

1.2 Research Questions

This work focuses on the usage of fiducial markers. Therefore, different marker setups are investigated and compared to each other as well as to a common stereo method, represented by Semi-Global-Matching (Hirschmüller, 2007, SGM). In this context, the following research questions are explored:

(1) Which configurations of fiducial markers are suitable for estimating the relative distance?
Different markers are discussed, selected and applied in various configurations. They are compared to each other and to a SGM approach based on real-world and simulated data.

(2) How are the individual methods influenced by uncertainty of input parameters?
A Monte-Carlo-Simulation setup is used to estimate the uncertainty of each method and the correlation to geometrical calibration parameters.

(3) Can the result of marker-based methods be improved, when applied in a stereo system?
All individual estimations in the stereo setup are collected and combined based on their estimated uncertainty from the Monte-Carlo-Simulation.

This work compares and evaluates the investigated methods based on the quality of the estimated distance. The required computational time is considered secondary, since the implementations of the individual methods differ in their expenditure in run-time optimization.

1.3 Organization of the Thesis

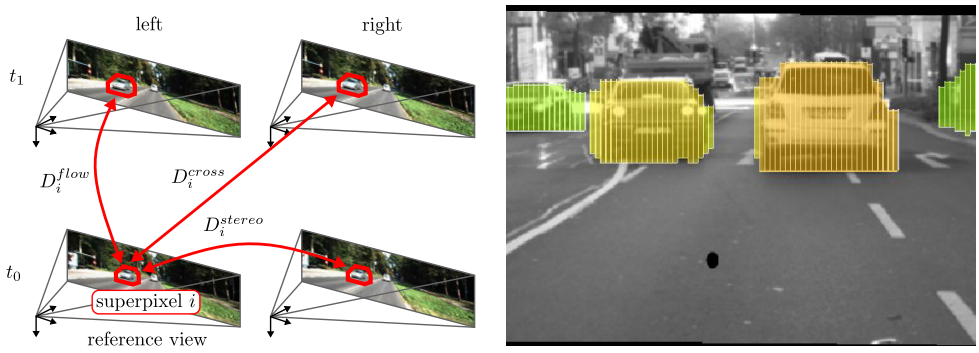
This thesis is organized as follows: Chapter 2 reviews state-of-the-art methods available in the field of relative positioning of autonomous vehicle (further abbreviated by RPV) based on monocular and stereo camera -based systems.
Chapter 3 describes the used camera model, sensor characterization and calibration. Moreover, underlying methods for detecting fiducial features that are selected in chapter 2 and for estimating the position are recapitulated. Also, the basics of image synthesis are outlined since it forms the basis for the following chapter.
In Chapter 4 the proposed evaluation pipeline is presented. This includes a more detailed explanation of the simulation setup, which is already used to substantiate various design decisions of the following section, as well as a presentation of the used real-world datasets.
The applications of the RPV-methods are discussed in Chapter 5. This includes presenting different marker configurations and how they are applied for estimating the distance in different RPV-methods. A short preliminary evaluation is attached to each new presented method, based on the previously introduced evaluation pipeline.
The presented RPV-methods are then evaluated in Chapter 6 with particular reference to the established research questions of Section 1.2.
Finally, the results are summarized and possible future work is discussed.

2 Related Work

Relative Positioning has been the subject of research for many years and has been investigated in many studies with different approaches. (Ponte Muller, 2017) presents a comprehensive review of different vehicle-based sensors to estimate the relative distance. That includes radar, LIDAR and monocular-, stereo- and time-of-flight camera systems. Furthermore, different cooperative methods are discussed that include absolute positioning methods and direct communication of autonomous vehicles. Comprehensive reviews for vision based vehicle detection and distance estimation are presented in (Bernini et al., 2014; Dhanaekaran et al., 2015; Sivaraman and Trivedi, 2013b).
The following survey of related work differentiates between stereo-based and monocular approaches. In conclusion, the methods selected for this work are named and justified.

2.1 Stereo Methods

Distance estimation based on stereo-camera system relies on assigning correspondences between two image points in both images. Afterwards triangulation is applied to estimate the distance of each match, explained in Section 3.2.1. Consequently, the quality of the estimated distance depends on the correctness and accuracy of the matches. Matching algorithms are classified as feature-based or area-based matching methods (Remondino et al., 2013) and calculate costs for each match candidate to solve the correspondence problem. It is distinguished in local (winner takes all strategy) and global methods (with global reasoning), which represent the balancing between accuracy and effort. A good trade-off is implemented with SGM (Hirschmuller, 2005), which forms the foundation of many stereo-based Advanced Driver Assistance Systems applications. It combines several one-dimensional optimizations from multiple directions to estimate the disparity of one pixel. This results in a computational efficient disparity map computation, while the accuracy is comparable to the result of global methods. Also, SGM can be extended for specific tasks. For instance (Hermann and Klette, 2013) proposed an iterative SGM approach that stabilizes road surfaces on challenging data by reducing the search space using pre-evaluated disparity priors saved in a semi-global distance map.
Such stereo measurements are represented in different forms to detect and reliably estimate the distance to various objects. (Elfes, 1989) proposed occupancy grid mapping, a probabilistic occupancy map which represents the world as a rigid grid of cells with the possible states: free, occupied, unknown. (Badino et al., 2007) introduced the concept of the polar occupancy grid map, which was developed further in (Badino et al., 2009) by stixels. By applying a subsequent free space computation and background-foreground mapping to calculate the height and the base point of the stixels, the depth is obtained with high accuracy. The stereo measurements of this approach and further developments (Cordts et al., 2017; Erbs et al., 2011) are based on SGM. A related representation is the Digital Evaluation Map (DEM), which represents the measurements in a height-based occupancy grid (Oniga and Nedevschi, 2010). It has been applied for obstacle detection and road surface estimation. Tracking of vehicles has been accomplished with the dynamic DEM (Danescu and Nedevschi, 2014).



(a) Superpixel (Menze and Geiger, 2015) (b) Stixel (Erbs et al., 2011)

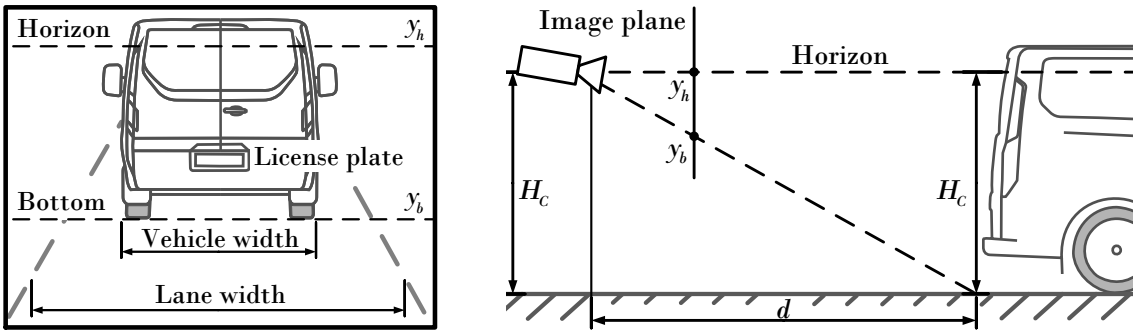
Figure 2: Visualization of exemplary stereo-based methods

Another widespread method is Optical flow (Lucas and Kanade, 1981) that is applied to stereo approaches to make use of temporal information. For instance (Lenz et al., 2011) matches interest points in a temporal as well as stereo sense to distinguish between moving and rigid spatial objects. (Menze and Geiger, 2015) proposed a slanted-plain model assuming that the 3D structure of the scene can be approximated by a set of piece-wise planar superpixel (Figure 2 (a)). They optimized their observation by using a disparity map generated by SGM and a CAD model to apply 3D model fitting.

2.2 Monocular Methods

While stereo-cameras can directly estimate the 3D coordinates of an object, monocular systems require prior knowledge about the size of the object. This prior knowledge can be attached to natural features, such as the gauge width or the vehicle height, or to fiducial features in the sense of markers of known size.

A selection of natural features is illustrated in Figure 3. (Stein et al., 2003) describes a vision-based Automatic Cruise Control system that uses the known extrinsic calibration of the observing camera. By assuming a straight surface, they estimate the distance from the horizontal line and the detected bottom of the vehicle by the intercept theorem, illustrated in Figure 3 (b). This concept is refined in the work of the patent document (Stein et al., 2012). For triangulation the width of the vehicle is used which itself is estimated and updated over several successive frames. To do so, they consider movement of the horizon, occurrence of bumps and movement of lane marks, all combined in an energy function. Similarly, (Nakamura et al., 2013) proposed a sequential Bayesian framework to update the unknown vehicle width based on a frame-based width estimation and tracking of the vehicle by using a Kalman Filter (Thrun et al., 2006, p.34). The problem of the vehicle‘s own pitching is especially treated in (Park and Hwang, 2014) in the context of a forward collision warning system. (Lessmann et al., 2016) fuses vehicle width information obtained by a vehicle classifier and addtional ground plane information from a lane detection system. These methods show the potential of monocular camera systems using natural features. However, these methods employ assumptions about the environment whose inaccuracies can lead to inadequate reliability in the context of safety-critical applications.

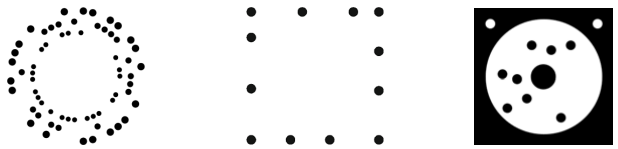


(a) Captured image (b) Triangulation geometry

Figure 3: Visualization of natural features, based on (Park and Hwang, 2014). H_C represents the calibrated height of the monocular camera. y_h and y_b illustrate the projections of the horizon and vehicle bottom in the image plane.

In contrast, license plate based distance estimation employs prior knowledge of a pattern with fixed dimension. License plate recognition has been already applied in (Chen and Chen, 2011) and in (Lu et al., 2011), which additionally uses the vehicles taillight to recognize the license plate. Recent research (Liu et al., 2017) proposed a robust license plate detection based on a fusion method and estimated the distance based on the known plate height to avoid influences caused by turning-vehicles. These applications present a reliable distance estimation when attaching a pattern of known size to the vehicle. However, this thesis is rather addressed to train applications, where license plates are not present. Nevertheless, this method demonstrates the opportunities that come with vehicle-attached markers of known size.

Fiducial features (also referred as tags) exist in various appearances and are applied to different applications such as 3D reconstruction, pose estimation and object identification. Therefore, they generally consist of a rectangular or circular, two-dimensional shape of known dimension and include a visual code to identify the tag. The following summary starts with a brief overview of tags created for reconstruction and ends with markers especially designed for pose and distance estimation.
For reconstruction an accurate pose estimation of the camera is presupposed that requires multiple well extracted image features. A state-of-the-art tag is Rune-Tag (Bergamasco et al., 2011, 2016). It is based on multiple small black circles likewise arranged in circles, shown in Figure 4 (a). By accommodating redundant coding, they achieve great robustness to occlusion and a large number of dots favor an accurate pose estimation. Similarly, (Bergamasco et al., 2013) proposed Pi-Tag which combines multiple circles in a rectangular shape. By exploiting collinearity and cross-ratios



(a) Rune-Tag (b) Pi-Tag (c) X-Tag

Figure 4: Fiducial markers used for reconstruction

(a) ARToolKit (b) ARTag (c) AprilTag (d) Chroma-Tag

Figure 5: Selection of rectangular shaped fiducial markers

they reduce the influence of perspective distortion. (Birdal et al., 2016) proposed X-Tag which uses a circular shape with randomly positioned inner dots and two white additional dots to identify the markers orientation. While using multiple tags in a non-planar configuration, they show superiority over using co-planar circle features given by Rune-Tag during 3D reconstruction. Even though these tags allow precise pose estimation they need many pixels to be detected, which is why they are not well suited for distance estimation.

Rectangular shaped markers are frequently applied for pose estimation. A selection is presented in Figure 5. The outer rectangular shape facilitates a reliable recognition and provides four points that are used for the Perspective-n-Point problem (PnP) to estimate the camera pose. Early approaches are represented by ARToolKit (Kato and Billinghurst, 1999) and (Ababsa and Mallem, 2004), both originally developed for real-time augmented reality systems. However, ARToolKit was successfully applied in (Seng et al., 2013) to estimate the pose of an unmanned aerial vehicle. To differentiate between multiple markers the inner area of the tag is usually equipped with a coding system. While ARToolKit uses Latin characters that are disadvantageous due to their high computational effort of decoding, ARTag (Fiala, 2005) is equipped with a binary coding system based on forward error correction, which leads to easier generation and correlation of tags. Furthermore, this tag is robust to changes of lightning and partially occlusion. (Olson, 2011) proposed AprilTag which improves upon ARTag in detection and encoding by using a graph-based clustering for detecting the tag borders and a new coding system preferring complex pattern to reduce the false positive rate. AprilTag is fully opensource and has been successfully used in many applications. In (Britto et al., 2015) AprilTag are used to estimate the pose of an unmanned underwater vehicle and (Winkens and Paulus, 2017) applied these markers for truck tracking in short range before building a model of natural features for long range tracking. (Wang and Olson, 2016) improved AprilTag especially for the reliable detection of small tags. (Walters and Manja, 2015) proposed Chroma-Tag and expanded the coding system to use color information to provide more distinguished IDs. (Mangelson et al., 2016) proved the sensibility of AprilTag to image exposure experimentally and expanded AprilTag with

(a) Intersense (b) Cucci (c) Calvet (d) WhyCon (e) WhyCode

Figure 6: Selection of circular shaped fiducial markers

multiple circles for robustness. (Pertile et al., 2015) evaluated the uncertainty of a vision system with a rectangular marker based on a Monte-Carlo-Simulation.
Circle-based features are often applied since their image projections are cheaply and robustly detected. A selection is presented in Figure 6. (Naimark and Foxlin, 2002) used the centers of four of their depicted markers for pose estimation in the context of visual-inertial self-tracking. Similarly, (Wilson et al., 2014) used four LED-markers on a plane for applying PnP for a formation flight. (Krajník et al., 2013) proposed WhyCon, a fiducial marker based on a simple concentric contrasting circle (Gatrell and Hoff, 1991). WhyCon impresses with short detection time, long detection range and precise pose estimation based on one marker. It is mainly applied for tracking multiple mobile robots and is fully opensource. (Lightbody et al., 2017) proposed WhyCode that extends WhyCon with a circular binary code. (Calvet et al., 2016) investigated an opensource fiducial marker based on multiple concentric circles under challenging conditions as motion blur and weak lightning. (Cucci, 2016) proposed a circular fiducial feature design based on two black circles coded with white blobs for aerial photogrammetry. The hierarchical design allows a reliable detection while landing and from far distances.

2.3 Outline

The presented related work shows the intensity of research related to relative distance estimation of autonomous vehicles. Natural features have shown to be an encouraging solution since only one camera is required and no further attachments to the preceding vehicle. However, disadvantageous is the potential for inaccurate or even wrong assumptions and the general non-public provision of implementations. The implementations of fiducial feature detection on the other hand are mostly public available, which makes them attractive for further research. Furthermore, the known dimensions of fiducial markers prevent the system from inaccurate assumptions that are crucial in safety-critical applications such as dynamic train composition. As a consequence, this work focuses on the application of fiducial markers. Two tags are selected. First, AprilTag is chosen because of its popularity and frequent use. Second, WhyCon is used since its simplicity promises a long range application.
Many stereo-based approaches rely on SGM. Therefore, this work evaluates the direct distance estimation by SGM without any further extensions.

Several benchmarks evaluating and comparing different vision-based methods for vehicle applications are publicly available. (Geiger et al., 2012) provides the popular KITTI-benchmark, which includes stereo-records completed with CAN-Bus and LIDAR data. (Menze and Geiger, 2015) partially extended this dataset by providing ground truth labeling of vehicles especially for scene flow applications. (Sivaraman and Trivedi, 2013b) and (Caraffi et al., 2012) provide image sequences of a monocular camera for vehicle detection and tracking. However, these benchmarks are not applicable in this work since different fiducial markers in various configurations will be applied. Therefore, two task-specific benchmarks are presented in Section 4.2.

3 Fundamentals

This chapter describes basic theories and methods from the field of computer vision and computer graphics that are used in this work. First, the geometry of single and stereo camera views is stated in conjunction with utilized mathematical models for describing camera characteristics. Second, methods are introduced that allow to estimate the distances to objects projected onto the image plane. Third, the methods selected in the previous section are recapitulated to call attention to important characteristics. Finally, the underlying rendering pipeline is explained, which is used to extract synthetic image data within this work.

3.1 Camera Modeling

This chapter introduces mathematical models that are used in this work and illustrates corresponding parameters that are either used in the simulation for variation of the rendered scene or in the application stage to model uncertainty (see Section 4).

3.1.1 Camera Model

To model the projection of a camera, the pinhole camera model (Schreer, 2005, p.40) is used, illustrated in Figure 7 (left). It describes the central projection of an object point M onto the image plane Π, which is defined parallel to the xy-plane of the camera coordinate-frame CL and is placed in front of the camera in the mathematical model. Related, the principal point c describes the point on the image plane that is the intersection of the principal axis z and Π. The projected image point m is then defined in image coordinates $(u, v)^T$.

$$P = K * P_N = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \alpha & v_0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \text{with } \alpha = f/\delta \tag{1}$$

Direct mapping of an object point from camera coordinates to ideal image coordinates is realized by the projection matrix P of Equation 1, which consists of the camera

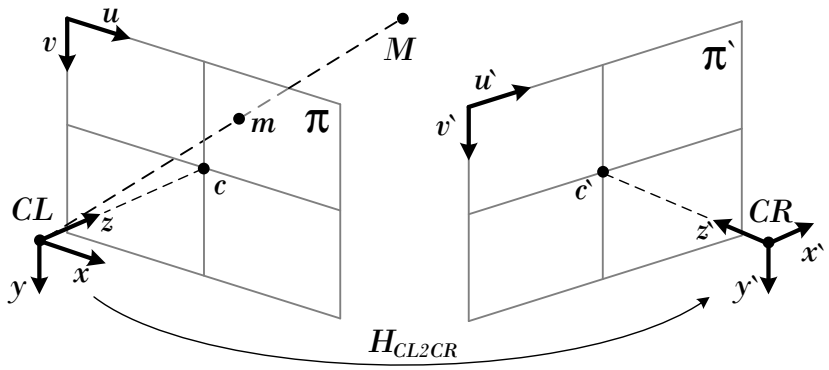


Figure 7: Camera model in a stereo setup, based on (Griesbach et al., 2014)

matrix K and a normalized 3x4 matrix. The algebraic model of K is composed of the principal point $c = (u_0, v_0)^T$ and the principal distance in pixel units α, which is based on the focal length f and pixel size δ.

$$wm = P * H_w * M \tag{2}$$

In the case that M is defined in a world-coordinate frame, it firstly is transformed into the camera-coordinate frame CL by an associated Euclidean homography matrix H_w before it is applied to the projection matrix, as shown in Equation 3. w is a scale factor for the transformation into the two-dimensional Euclidean space.

$$w`m` = P * H_{CL2CR} * H_w * M \tag{3}$$

In the case of a stereo setup, as illustrated in Figure 7, the object point is additionally projected onto the right image plane Π` by Equation 3. It applies the homography matrix H_{CL2CR} after the world-to-camera frame transformation H_w. Thus, H_{CL2CR} describes the transformation from the left to the right camera-coordinate frame.

3.1.2 Distortion Model

When using real lenses, deviations to the ideal pinhole model occur as for example in forms of defocus, spherical and chromatic aberration, coma, and image distortion, which is generally most significantly. For illustrative purposes, Figure 8 shows two exaggerated forms of radial distortion.
(Brown, 1971) proposed a distortion model that models radial distortion δ_r and tangential distortion δ_t, as formulated in Equation 4. It describes the relation between the distorted point $(\hat{u}, \hat{v})^T$ and the ideal point $(u, v)^T$. This model is frequently used in many applications such as (Heikkila and Silven, 1997; Zhang, 2000).

$$g(u,v) = \begin{pmatrix}\hat{u}\\ \hat{v}\end{pmatrix} = \begin{pmatrix}u\\ v\end{pmatrix} + \delta_r(u, v, k_{1,2,3}) + \delta_t(u, v, p_{1,2}), \qquad \text{with } r^2 = u^2 + v^2 \tag{4a}$$

$$\textit{and} \qquad \delta_r(u, v, k_{1,2,3}) = \begin{pmatrix}u\\ v\end{pmatrix} * (k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{4b}$$

$$\textit{and} \qquad \delta_t(u, v, p_{1,2}) = \begin{pmatrix}p_1(3u^2 + v^2) + 2p_2 xy\\ p_2(u^2 + 3v^2) + 2p_1 uv\end{pmatrix} \tag{4c}$$

The radial parameters k_1, k_2, k_3 and tangential parameters p_1, p_2 of this model are estimated using a calibration process. The calibration estimates a probabilistic distribution for each parameter by assigning a Gaussian distribution with a bias and a standard deviation. The standard deviation describes the uncertainties of the individual parameters, which depend among others on the accuracy and the number of detected chessboard corners in the image used for the calibration. Depending on the manufacturing quality of the camera to be calibrated, single parameters of the model are usually set to zero. This improves the accuracy and uncertainty due to fewer parameters while the number of used image points remains unchanged. Thus,

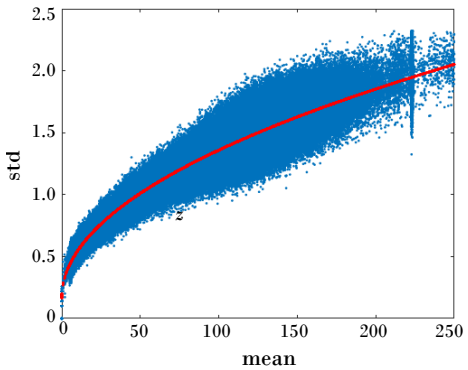
(a) Pincushion distortion (b) No distortion (c) Barrel distortion

Figure 8: Visualization of the essential forms of radial distortion

the tangential parameters are defined as zero within this work since the used stereo camera [DLRStereo] has cameras with negligible tangential distortion.

3.1.3 Noise Model

In addition to degradation caused by the lens, the conversion of the captured light into a digital signal adds noise to the image. Image noise sources are mainly classified in fixed pattern noise and dynamic noise. Fixed pattern noise such as Photo Response Non Uniformity (PRNU) and Dark Signal Non Uniformity (DSNU) are usually automatically corrected by the camera itself. In contrast, dynamic noise varies between each captured frame due to read-out noise and photon noise. A comprehensive review of different noise models is presented by (Boyat and Joshi, 2015). A recent approach is proposed by (Zhang et al., 2017), which is used in this work to degrade the image. Figure 9 shows the root-shaped dependency of noise's standard deviation to the pixel grey-scale values. This distribution is described by Equation 5. As formulated in (Zhang et al., 2017), it firstly composes of a parameter N_E representing the electronic noise of the camera.It ensures a standard deviation greater than zero even for dark pixels. And second, the the grey value I divided through a gain parameter G represents shot noise. The shown image noise of Figure 9 is generated by the simulator based on the calibrated parameters $N_E = 0.2658$ and $G = 59.1944$.



$$\text{Noise} = \sqrt{N_E^2 + I/G} \qquad (5)$$

Figure 9: Illustration of the noise model (Zhang et al., 2017). Blue points show all standard deviation for each intensity. The red curve is fitted by Equation 5.

3.2 Position estimation

This chapter outlines methods for estimating the position of the camera with respect to another object. Three different methods are introduced which work according to different principals.

3.2.1 Stereo Triangulation

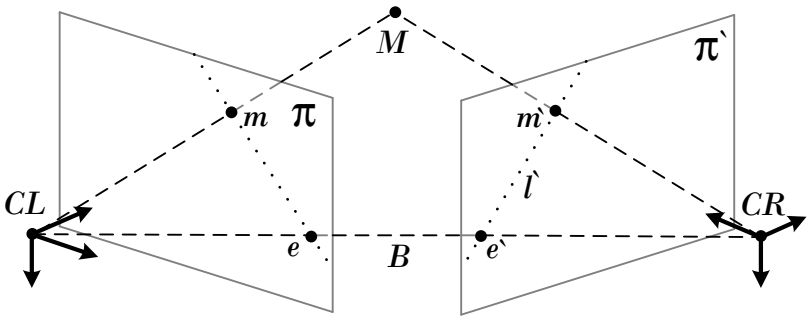
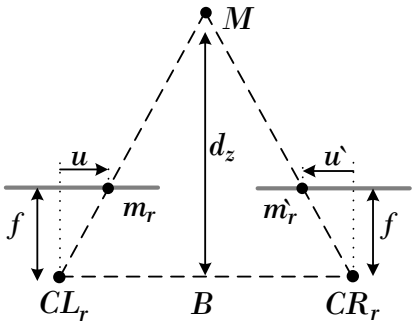


Figure 10: Epipolar geometry, based on (Schreer, 2005, p.69)

The estimation of the distance to an object point M by its projected image points m and m‘ in a stereo setup is based on the epipolar geometry, illustrated in Figure 10. The baseline B describes the connection of the origins of the camera-coordinate frames CL and CR. Its intersections with the image planes Π and Π’ define the epipols e and e’. Related, the object point M and the origins of CL and CR define the epipolar plane, while m, m’, e, e’ lie on this plane. Its intersections with the image planes denote the epipolar lines. The epipolar geometry then states that the related image point of m in image plane Π’ lies on the epipolar line l’.
Thus, the epipolar geometry reduces the costs of matching image features due to a smaller search space. In the case of a known stereo geometry H_{CL2CR}, rectification is used to simplify the epipolar geometry (Schreer, 2005, p.105). By virtually rotating CL and CR to form an axis-parallel camera system, the epipolar lines become parallel. Consequently, the corresponding image point m_r’ lies in one image row of Π’$_r$. The result of the rectification is the normalized stereo case shown in Figure 11, which reveals an intercept theorem for estimating the depth d_z of M in rectified camera



$$d_z = \frac{B * f}{\delta * (u - u`)} \tag{6}$$

Figure 11: The stereo normal case, based on (Schreer, 2005, p.67)

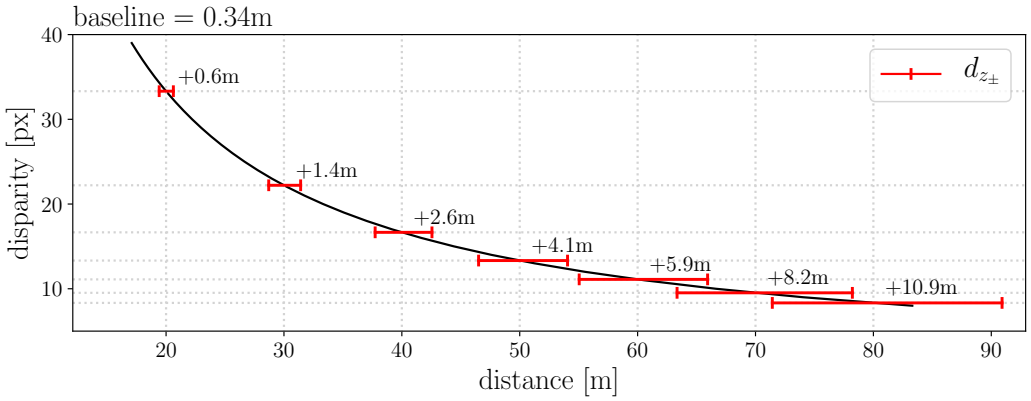


Figure 12: Theoretical dependency of disparity and distance in a stereo setup [DLRStereo] with a baseline B of 0.34m.

coordinates. The relation is described in Equation 6 in which δ states the pixel size. It reveals a strong dependency between the accuracy of the estimated disparity $(u - u')$ based on the image point matching and the resulting calculated distance d_z. Figure 12 illustrates this dependency. Red lines show the resulting distance deviations $d_{z_\pm}$ for $\pm$1px deviation in disparity space starting from the disparity at the displayed distances. E.g. the exact disparity value at 70m is 9.52px and at 80m it is 8.33px. The value of the positive deviation is displayed. It is recognizable that the dependency incarnates a square rising shape for smaller disparities and thus larger distances. In addition, a disparity deviation of +1px at 10m results in a distance deviation d_{z_+} of 0.15m.

3.2.2 Perspective-n-Point

The "Perspective-n-Point problem" (Fischler and Bolles, 1981) can be applied to estimate the pose and thus the distance between the camera with respect to another object if known correspondences between 3D world points and their 2D projections exist. A comprehensive overview and comparison is presented in (Urban et al., 2016).
Frequently used representatives are Efficient-PnP (Lepetit et al., 2009, EPnP) and Robust-PnP (Li et al., 2012, RPnP), which are tested for this work[1]. Both represent non-iterative linear solutions to the PnP problem. EPnP accomplishes linearity by expressing the 3D reference points as a weighted sum of four virtual control points and refining the solution using a Gauss-Newton optimization. RPnP on the other hand investigates 3-point subsets by exploring the local minima of the equation system, which is based on the fourth-order polynomials (Quan and Lan, 1999), in terms of least-squares residual. For each minimum the camera pose is estimated before the final pose is selected based on the reprojection error.
Figure 13 shows a comparison of both methods concerning their accuracy and robustness to noised image and reference points based on a Monte-Carlo test. In an arbitrary

[1]All methods used in the comparison of Figure 13 are accessible for this work as c++ implementations [OSVisionLib], while EPnP and iterative PnP are implemented by [OpenCV]

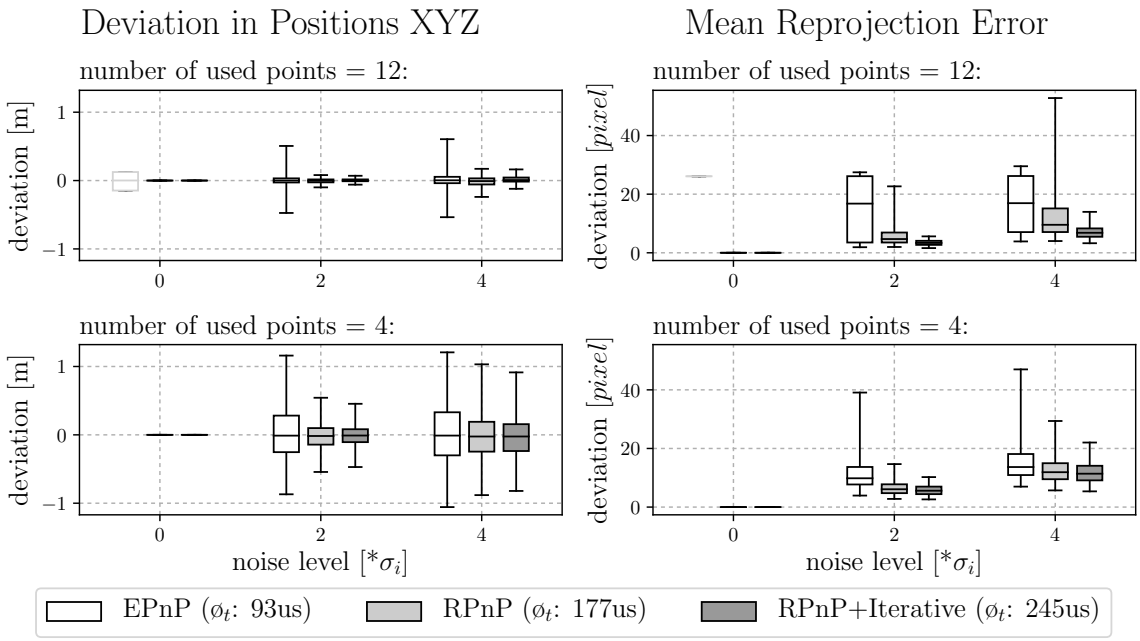


Figure 13: Comparison of PnP-methods in a planar setup, based on position deviation and reprojection error of noisy image- and world point correspondences

scene[2], which simulates the projection of the corners of three coplanar AprilTags, four and twelve correspondences are used to solve the PnP problem. Each method is applied to different noise levels nl, which applies Gaussian noise to the position of the reference points by $\sigma_{3D} = nl * 1\text{cm}$ and to image points by $\sigma_{2D} = nl * 1\text{pixel}$. Each distribution of Figure 13 is based on 100.000 iterations in the Monte-Carlo test. For a description of the box plots, please visit Section 4.4 (p.28).
Figure 13 (left) shows the deviation of the estimated homography to the ground truth matrix based on its translation, while each box includes the deviation in all three directions. Concerning an increasing noise level RPnP is more accurate using either four and twelve points. The same result is maintained when comparing the mean reprojection error. The grey-lined box that marks the result of EPnP at zero noise indicates obvious non-valid calculations, which implies a leak in robustness by EPnP.
In this work PnP is mainly applied for four correspondences provided by one AprilTag or four WhyCon, which implies the application of RPnP because of its superior performance in this case. The result is then refined by applying an iterative PnP (Levenberg-Marquardt, [OpenCV]) approach that directly optimizes the pose based on the reprojection error. Figure 13 shows a clear improvement with respect to stand-alone EPnP or RPnP. However, the quality and validity of the iterative approach strongly depends on the initial pose assumption. For this reason, the approach RPnP+Iterative is used in the further course of the work, names as PnP.

[2]A visualization of this scene with coplanar 3D reference points can be found in Appendix A.1. Also a scene with non-planar reference points is provided.

3.2.3 3D Position Calculation of a Circle

The projected ellipse of a circular shaped marker can be used to estimate its pose, detailed in (Krajník et al., 2014, p.9)[3]. In the further course of this work, this method is applied to estimate the position of the circle of a WhyCon marker (WhyCon+Circle). The projected ellipse is defined by its center $\boldsymbol{c_e}$ (u, v) and semiaxis $\boldsymbol{e_0}, \boldsymbol{e_1}$, which result from the marker detection of Chapter 3.3.2. It is transformed to a canonical camera system to compensate the radial distortion at the position of the detected ellipse. Illustrated in Figure 14 the image coordinates of the ellipses vertices $\boldsymbol{a_{0,1}}$ and $\boldsymbol{b_{0,1}}$ are calculated and un-distorted by using the model of Section 3.1.3. The transformed image points $\boldsymbol{a'_{0,1}}$ and $\boldsymbol{b'_{0,1}}$ are used to define the the transformed ellipse, resulting in $\boldsymbol{c'_e}$ (u'_c, v'_c) and $\boldsymbol{e'_0}, \boldsymbol{e'_1}$.

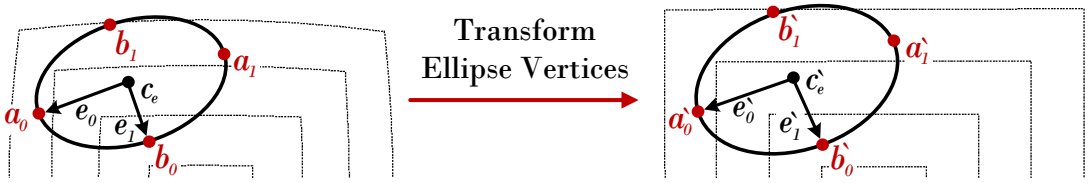


Figure 14: Visualization of the concept to compensate radial distortion of ellipses

The resulting parameters are then used to establish the parameters of the conic, defined in the ellipse characteristic equation of Equation 7.

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}^T \begin{bmatrix} q_a & q_b & q_d \\ q_b & q_c & q_e \\ q_d & q_e & q_f \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = 0, \quad \text{with} \quad \begin{cases} q_a = +e'_{0u}e'_{0u}/|\boldsymbol{e'_0}|^2 + e'_{0v}e'_{0v}/|\boldsymbol{e'_1}|^2 \\ q_b = +e'_{0u}e'_{0v}/|\boldsymbol{e'_0}|^2 + e'_{0u}e'_{0v}/|\boldsymbol{e'_1}|^2 \\ q_c = +e'_{0u}e'_{0u}/|\boldsymbol{e'_1}|^2 + e'_{0v}e'_{0v}/|\boldsymbol{e'_0}|^2 \\ q_d = -u'_c q_a - v'_c q_b \\ q_e = -u'_c q_b - v'_c q_c \\ q_f = +q_a {u'_c}^2 + q_c {v'_c}^2 + 2q_b u'_c v'_c - 1 \end{cases} \tag{7}$$

By means of an eigenvalue analysis of the conic with the eigenvalues $\lambda_0, \lambda_1, \lambda_2$ and eigenvectors $\boldsymbol{q_0}, \boldsymbol{q_1}, \boldsymbol{q_2}$, the position $\boldsymbol{x_c}$ of the pattern with the diameter d_0 is estimated by Equation 8.

$$\boldsymbol{x_c} = \pm \frac{d_o}{\sqrt{-\lambda_0 \lambda_2}} \left(\boldsymbol{q_0} \lambda_2 \sqrt{\frac{\lambda_0 - \lambda_1}{\lambda_0 - \lambda_2}} + \boldsymbol{q_2} \lambda_0 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_0 - \lambda_2}} \right) \tag{8}$$

[3]The implementation is provided in the opensource package[WhyConLib].

3.3 Image Processing

This chapter considers fundamental methods selected for application in Chapter 2 in more detail. This includes a short summary and outline of the methods characteristics.

3.3.1 AprilTag

AprilTag is a "visual fiducial system that uses a 2D bar code style" (Olson, 2011), shown in Figure 5 (c, p.6). Its detection is divided into two main steps: the detection of the pattern and the coding system used to identify the different tags.
The detection phase starts by smoothing the image to reduce the influence of image noise. Then for each pixel the gradient and its magnitude are estimated. The use of gradients reduce the influence of exposure. Based on the gradients, lines are detected by a graph-based clustering method. The set of detected lines is then investigated by a recursive "depth-first search with a depth of four" (Olson, 2011, p.4) to find quads of lines. For each candidate quad the 2D homography transformation from the tag coordinate system into the system defined by the four corners of the quad is estimated. This homography is then used to calculate the image position of each bit to extract the pixel values. Finally, a proposed spatially-varying threshold method is used to classify black and white values, which also increases the robustness to exposure.
The extracted binary codes of the candidate quads are used to distinguish between patterns with different IDs and to filter out quads with invalid IDs. Therefore, AprilTag consists of a complex coding system. First, it rejects code words that result in simple geometric patterns under the assumption that complex patterns occur less frequently in nature. Second, tag clusters are chosen in such way that the entropy is maximum in each bit by maximizing the Hamming distance.
The encoding and identification of the binary code plays an important role in the AprilTag pattern detection. The quad detection generates many candidates, which are verified by the subsequent identification[4]. That means the maximum detection distance of a AprilTag is bounded by the granularity of the included code.
The result of the Apriltag pattern detection are the four corner points of each tag, which are used to apply a PnP approach, described in Section 3.2.2.

3.3.2 WhyCon and WhyCode

The detection of the circular pattern of Figure 6 (d, p.6) is based on the assumption of a coherent circular white segment enclosed by a ring-shaped black segment. The algorthm is proposed and described in detail in (Krajník et al., 2013, 2014).
Given a new frame *Image* a buffer *pixel_class* is initialized, which is used to store the information for each pixel whether it is black, white or initially unknown. Then, Algorithm 1 is applied multiple times to find the next pattern until all possible patterns in the image are detected or a maximum number is reached. The algorithm starts at a passed pixel position p_0 and iterates pixel by pixel i through the image until a black pixel is reached, classified by the passed threshold τ. Using a Floodfill algorithm,

[4] An example is provided in Appendix A.2.

all connected black pixels are segmented and additionally marked in the buffer. If the segment c_{outer} surpasses a minimum size and passes a simple roundness test to guarantee a circular shape, a new segmentation of white pixels is started from the center of c_{outer} to verify its annularity. The white segment c_{inner} is also investigated on its minimum size and roundness. Given both segments and prior knowledge about the proportion of the patterns inner and outer radius, a ratio test on the segments size is performed. Using all included pixels of the segments, the ellipses center (u_c,v_c) is defined by their mean position. Formulated in Equation 9, the ellipses semiaxis e_0, e_1 are calculated by the eigenvalues λ_0, λ_1 and eigenvectors v_0, v_1 of the covariance matrix C.

$$e_{0,1} = 2\lambda^{1/2}{}_{0,1}v_{0,1} \quad \text{with} \quad C = \frac{1}{s}\sum_{i=0}^{s-1}\begin{bmatrix} u_iu_i & u_iv_i \\ u_iv_i & v_iv_i \end{bmatrix} - \begin{bmatrix} u_cu_c & u_cv_c \\ u_cv_c & v_cv_c \end{bmatrix} \tag{9}$$

The resulting ellipse is subjected to a final test to verify its circularity.
For speeding up the detection process, the center of a detected pattern is stored and used in the next frame as start position p_0. Assuming that the projected pattern moves slowly in the image, the previous pattern center still lays on the projected pattern in the new frame. Thus, the pattern can be detected very fast in a stream of successive frames as shown in (Krajník et al., 2013).
In addition, the threshold used for the current frame is estimated while processing the previous frame by using the means of the patterns projected black μ_{outer} and white

Algorithm 1: WhyCon pattern detection, based on (Krajník et al., 2013, p. 6)

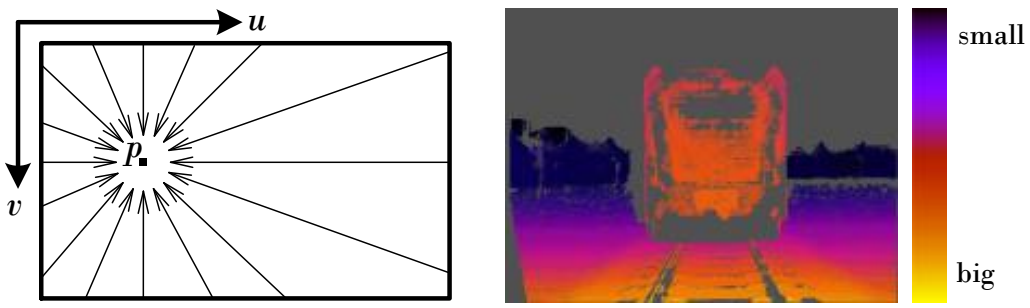
Data: p_0 - start position, τ - threshold, Image - current frame
Result: p_0 - next start position, τ - updated threshold, c - pattern data

```
i ← p_0 repeat
    if pixel_class[i]= unknown and Image[i] < τ then
        pixel_class[i] ← black
        c_outer ← flood_fill_segment(i, black)                ▷ check outer segment
        if verify_segment(c_outer) then
            j ← center(c_outer)                               ▷ check inner segment
            c_inner ← flood_fill_segment(j, white)
            if verify_segment(c_inner) then
                if check_ratio(c_outer,c_inner) then
                    e_0, e_1 ← ellipse_semiaxis(c_outer)      ▷ check segment relation
                    if check_concentricity(e_0,e_1) then
                        τ ← (μ_outer + μ_inner)/2
                        c = c_outer
                        c.valid = true
                        break                                 ▷ segment found
    i ← (i + 1)mod sizeof(Image)
until i ≠ p_0
p_0 ← i
```

pixels μ_{inner}. This technique ensures optimal thresholding during the segmentation and thus an accurate estimation of the pattern borders. This is necessary since the position estimation based on a projected ellipse (see Section 3.2.3) is highly affected by the estimated pattern borders. Also, a correction of estimated semiaxis is proposed that takes the true ratio of the inner and outer circle into account (Krajník et al., 2014, p. 8). However, this time-dependent determination of the threshold assumes a constant exposure, which only applies for indoor applications.
By reducing precision of the roundness check for the inner circle, non-circular inner white segments can be applied to the pattern. As illustrated in Figure 6 (e, p.6), WhyCode (Lightbody et al., 2017) extends the pattern by applying a binary code to the inner circle. For code identification, they combine a "Necklace code" with Manchester Encoding (Forster, 2000), which provides rotation invariance and different IDs.

3.3.3 SGM

Semi-Global Matching "[...] uses a pixelwise, Mutal Information based matching cost for compensating radiometric differences of input images" (Hirschmüller, 2007). It is based on a known interior and relative calibration of a rectified stereo setup. This implies that the corresponding pixel in the other image is known to lie on the same image line. This knowledge is used to apply a local smoothness constraint. This is usually realized by calculating the matching costs of all possible disparities for each pixel p on that line, registered in a matrix and applying dynamic programming to find the path through the matrix which has minimal costs. However, neighboring pixels of contiguously lines often show irregular jumps in disparity (Moratto, 2013). SGM efficiently solves this problem by combining several one-dimensional optimization from all directions as illustrated in figure 15 (a). The result is a dense disparity image with sub-pixel estimation that provides sharp object boundaries and fine details.
Figure 15 (b) shows an exemplary colored disparity image based on a simulated image pair with a pictured train. Distant objects have small disparities and near rather large disparities, as it can be traced in Figure 11 (p.11) with the disparity $(u - u')$. The colored disparity visualization shows large gaps colored in grey, which arise in untextured areas. The implementation of (Ernst and Hirschmüller, 2008) is provided by the DLR for this work.



(a) Directions for cost aggregation (b) Visualization of disparity

Figure 15: Illustration of Semi-Global-Matching, based on (Ernst and Hirschmüller, 2008; Hirschmüller, 2007)

3.4 Image Synthesis

This section briefly explains the principles of image synthesis on a graphic processing unit (GPU) used to evaluate the different approaches. Subsequently, an extended rendering pipeline is reviewed that allows to directly simulate image degradation during the rendering process. Finally, methods to increase the image quality are discussed.

3.4.1 Basic Rendering

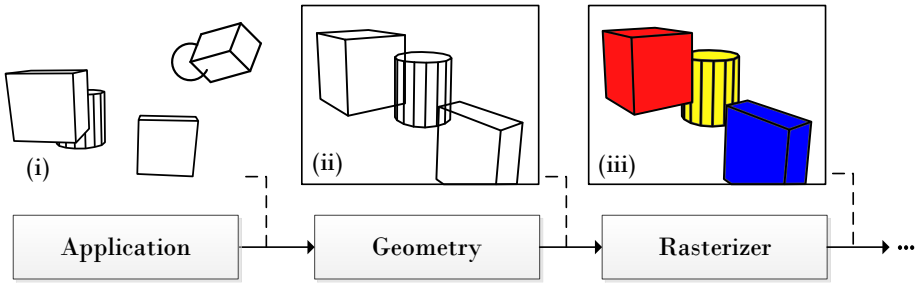


Figure 16: The graphic rendering pipeline

Synthetic data can be generated by using the graphic rendering pipeline. "The main function of the pipeline is to generate, or to render, a two-dimensional image, given a virtual camera, three-dimensional objects, light sources, shading equations, textures and more" (Akenine-Möller et al., 2008, p.11). This basic rendering pipeline consists of three conceptional steps, illustrated in Figure 16. First, the geometry of the scene is defined in the application step, which defines the positions of all elements based on the scene specification. Elements are points (or vertices), lines and faces. Faces are each defined by three vertices and represent the surface of the object. Second, all positions of the to be rendered objects are projected into image coordinates during the geometry stage based on a normalized camera model. Also the vertices that are not bordered by the image are clipped. Last, the rasterizer stage uses the transformed and projected vertices to compute the nearest face and set colors for each pixel defined by the face. The result is an ideal image that coincides with the pinhole model, as shown in Figure 16 (iii) or Figure 17 (ii).

3.4.2 Extended Rendering Pipeline

In this work, an extended graphics rendering pipeline is used, provided by the DLR (Lehmann, 2015, 2016). The extension is realized in two additional shader levels linked to the end of the basic shader pipeline and applies image degradation.
In the first step, lens distortion is realized in the *lens-shader*. It distorts the resulting image (Figure 17 (ii)) of the basic rendering pipeline by using the Brown distortion model, explained in Section 3.1.2 (p.9), and bilinear interpolation (Akenine-Möller et al., 2008, p.158). This is done by precomputing a lookup table on the central processing unit (CPU), which holds the position in the ideal image (ii) for each pixel of

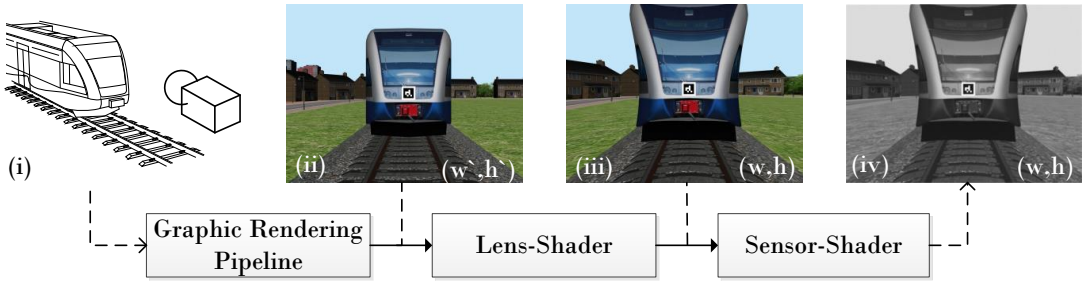


Figure 17: The extended rendering pipeline

the distorted image (iii). This lookup table is initially also used to adapt the camera model used in the basic rendering pipeline to increase its image borders (w',h') of (ii) in the way that all transformed positions of the distorted image can be mapped. This is necessary since the distortion can exceed the original image borders as shown in Figure 8 (p.10), which would lead to undefined regions in the distorted image (iii). Finally, various image degradation effects are modeled in the *sensor-shader*. This includes blurring with a Gaussian kernel, greyscaling, exposure and also an implementation of the noise model of Section 3.1.3 (p.10).

3.4.3 Anti-Aliasing

An important point to be considered when simulating photo-realistic visual data is Aliasing (Akenine-Möller et al., 2008, p.117). Different to image recording with real cameras, which integrates a variety of light beams for each image sensor, the standard rendering method in computer graphics is based on a single sampling of the scene for each pixel. This rastering of the scene leads to different image disturbances. The first row of Table 1 shows the two main disturbances while applying this rendering method. First, the textures of the pictured tags are under-sampled, which leads to conspicuous artifacts on the tags and pixel flickering when comparing the two successive frames. Second, the upper straight line of the wall border appears as a ragged edge (Jaggies). To prevent these image disturbances, three anti-aliasing methods are commonly applied.

First, supersampling samples several instances for each pixel, as shown in the visualization of supersampling in Table 1. This is achieved by rendering the scene in a higher resolution and subsequent downsampling to the original size. This method effectively removes all image disturbances as jaggies, artifacts and pixel flickering. However, the application of this method is limited to computing power and memory of the GPU. Second, multisampling samples several instances for pixels that are close to object edges as shown in the visualization of multisampling in Table 1. Thus, it is a specific optimization of supersampling. Its application to the exemplary frames shows that jaggies are effectively removed. Yet, artifacts and pixel flickering on the tags remain. Third, mip-mapping prevents undersampling of the textures. This method creates an image pyramid with different resolution levels of the texture and automatically determines the mip-map levels to use. The resulting color value is interpolated between the interpolated color values from the upper and lower resolution level, which in turn are estimated with bilinear interpolation of the neighboring pixel on each level. Table 1

shows that mip-mapping removes artifacts and pixel flickering within the tags, but jaggies on the wall border remain.

For this work, a combination of these three methods is used to ensure sufficient image quality. Supersampling is applied with a sampling grid of 4^2px for each original image pixel, followed by multisampling with a 4^2px grid for each supersampled pixel. Last, mip-mapping prevents undersampling of the pictured textures.

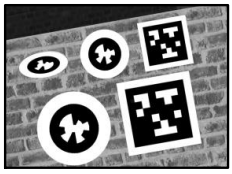
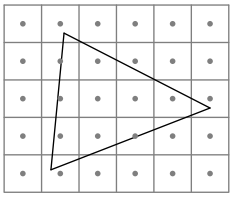
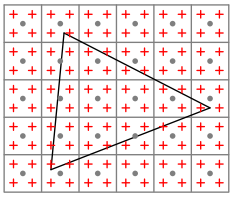
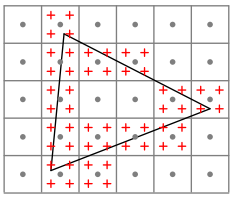
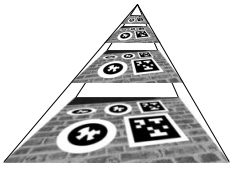
	Method	Visualization	Frame 1	Frame 2
(a)	Proposed Combination	 (Frame 1 Hi-Res) →		
(b)	Standard	 →		
(c)	Supersampling	 →		
(d)	Multisampling	 →		
(e)	Mip-Mapping	 →		

Table 1: Visualization of methods for anti-aliasing. Frames 1,2 represent low resolution images captured from a distance to the wall of 100m with a vertical camera shift of 0.05m from frame 1 to 2. The pictured AprilTag has a width of 0.6m. Frame 1 is shown in higher resolution in the visualization (a). Visualization (e) shows an exemplary image pyramid for mip-mapping. Visualizations (b,c,d) show the rastering of a scene with one face, the original sample positions as grey points and the new sample positions with red crosses with a sample grid of 2^2 for each original pixel, based on (Thoman, 2014).

4 Evaluation Pipeline

In this section an evaluation pipeline is proposed to evaluate and compare different RPV-methods. This setup allows to compare the accuracy of the applied distance-estimation methods under the influence of variation and uncertainty of versatile parameters such as exposure and calibration uncertainty. First, the general experiment setup and an outline of the stages are explained. Second, real-world experiment setups as well as the simulation concept are presented. Finally, the evaluation procedure is introduced.

4.1 General Setup and Definitions

The general setup of the experiments includes the recording of multiple frames from several static positions in front of the vehicle, respectively the train, as illustrated in Figure 18. The angle α and the distance d describe the considered parameters to set the camera position to the vehicle, defined by H_{CL2RP}. In the case of real-world experiments only d is considered. Additionally, the coordinate systems of the left camera CL, right camera CR, the reference point RP and of one exemplary tag is shown.
The proposed chain consists of three stages illustrated in Figure 19, which separate test-data aggregation, RPV-application and statistical evaluation. Within this chain the

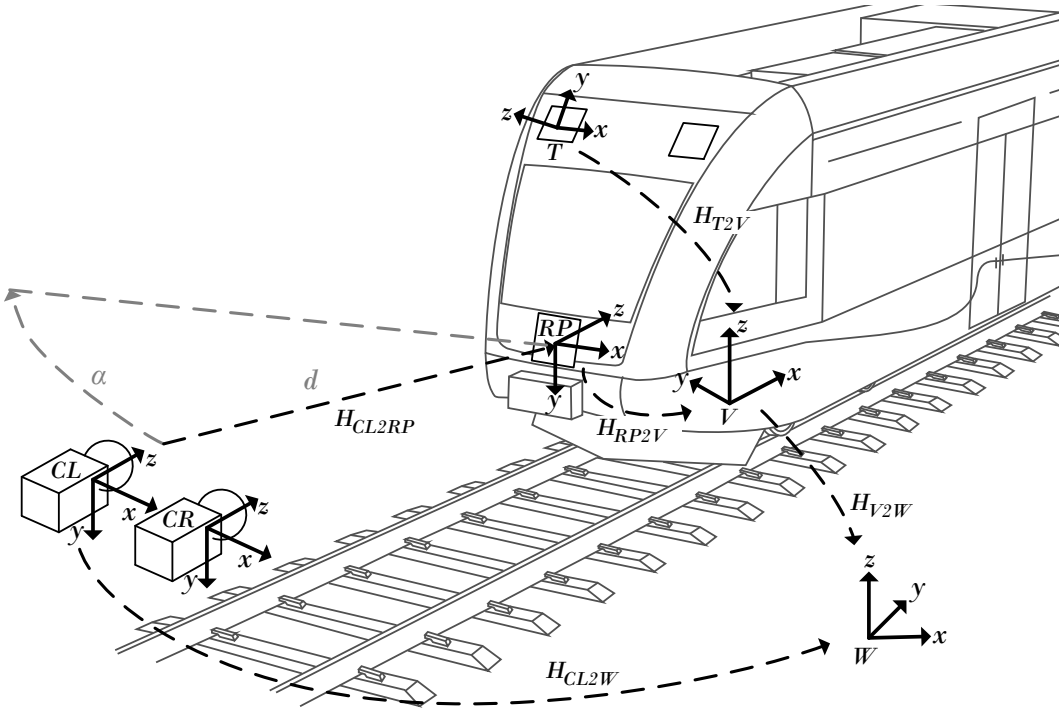


Figure 18: Specification of significant coordinate systems and transformations, e.g. the transformation of the vehicle V to the world coordinate system W is labeled with H_{V2W}.

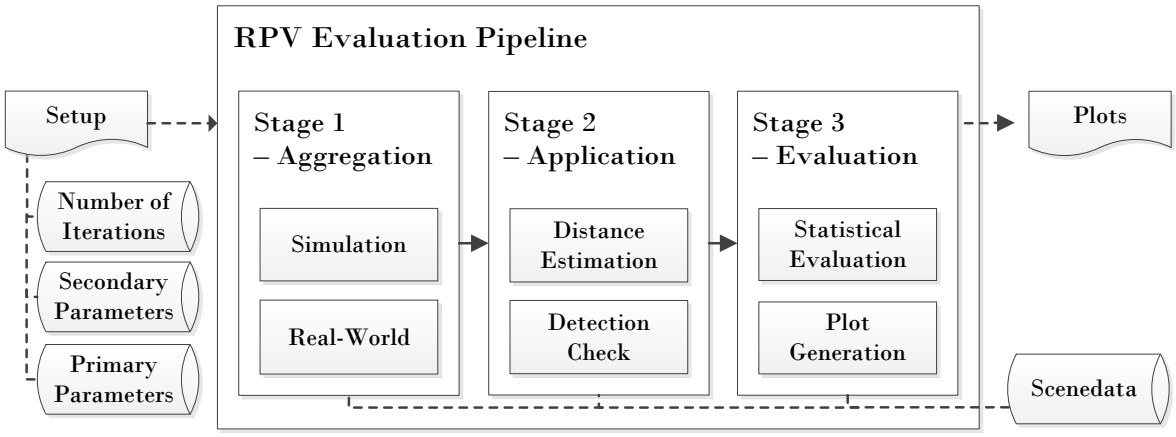


Figure 19: Schematic representation of the evaluation pipeline

data container *scenedata* is created and passed through, which holds the specifications of each individual iteration and is filled during each stage. In the first stage of the evaluation pipeline, the scenedata is set up using following parameter groups which are defined in the experiment setup:

- *Number of iterations* define how often each scene of the two-dimensional array of the scenedata is repeated with individual sampled *target* parameters.
- *Secondary* parameters define the Gaussian distributions of the variation and calibration parameters for the experiment. Based on this distribution the corresponding *target* parameters are sampled individually for each iteration.
- *Primary* parameters fix *target* parameters to specific values for each scene. For each simulated experiment up to two primary parameters are chosen which define the two dimensional scene-array of Figure 18 and 27 of the resulting plots.

As illustrated in Figure 20, each scene consists of multiple iterations, each defined by a set of parameters. These parameters are divided into four parameter groups:

- *Target* parameters hold the sampled input values that are used in the aggregation stage and the application stage such as the exposure or the noised focal length.
- *Ground truth* parameters hold the ground truth distance of the stereo-camera to the vehicle. In the case of simulated data, it also contains all true transformations of all simulated objects.
- *Support* parameters hold additional information about the scene, which would be estimated by another not-implemented algorithm. For instance, the labeling of the vehicle in the image used for the SGM approach in Section 5.3.1.
- *Estimated* parameters hold the estimated distance of the stereo-camera to the vehicle and information about the success of the marker detection.

During the aggregation stage and in the case of virtual test data, each iteration is simulated with the related *target* parameters and all required ground truth and support parameters are noted. In the case of real-world test data, the corresponding stereo

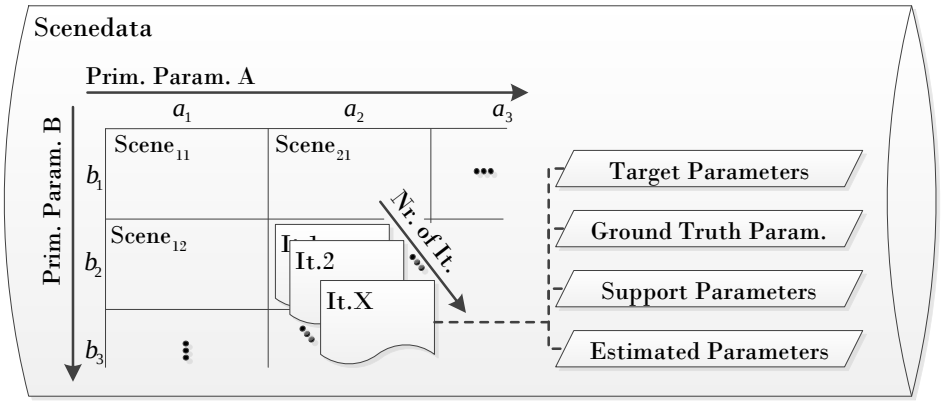


Figure 20: Illustration of the scenedata content

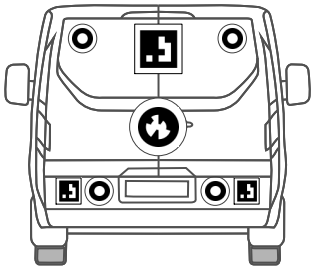
frames of the dataset are linked to each iteration and the ground truth and support parameters are defined by manually labeling the images.
In the application stage the individual methods are applied to each repetition to estimate the distance. The noised calibration parameters that are stored in the *target* parameters are applied in this stage. Afterwards, the detection of the markers in the image is checked by using the ground truth data to assess the success of the marker detection and thus the distance estimation. Finally, the generated data is evaluated and interpreted by the help of different plots in the last stage.

4.2 Real-World Datasets

To compare and validate the performance of the RPV-methods real-world benchmark datasets are indispensable. Since the focus of this work is the distance measurement to vehicles with attached fiducial markers, publicly accessible benchmarks are not usable. Due to this, two real-world benchmark datasets are proposed in this work. They consist of recordings of vehicles with attached AprilTags and WhyCon marker in various configurations taken from certain distances. Both conform to the setup of Figure 18. The recordings were taken in front of the vehicle ($\alpha = 0°$) in certain distances d. For each distance, multiple pictures were recorded that define the number of iterations of Figure 20. However, no secondary parameters were changed during the image captur-

(a) Type 1 (b) Type 2 (c) (d)

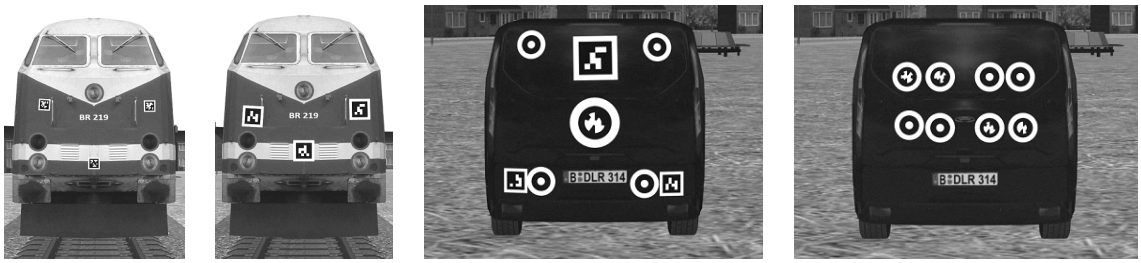
Figure 21: Exemplary left camera subimages from the proposed datasets. (Increased contrast and brightness for better illustration)



	Configuration	Occupied Area
(1)		$0.133m^2$
(2)		$0.135m^2$
(3)		$0.181m^2$
(4)		$0.215m^2$

Figure 22: Specification of the marker areas

ing. Thus, the only influence that varies between each iteration is image noise. For each position to the train or vehicle a reference measurement with a laser scanner [GLM-80] was conducted. Detailed information about the datasets can be found in Appendix B.3. The first dataset is based on a measurement campaign (Funk, 2017) in which two configurations of different AprilTags attached to a train (BR219) were recorded, each from three distances of up to 24m. Figure 21 (a) and (b) shows two exemplary subimages captured by the left camera. This dataset is used for a preliminary experiment to find a promising configuration for AprilTags. Charactaristic of this data is the visible overexposure, exemplary shown in Appendix B.3.1.
The second dataset is divided into two subsets. First, the setup of Figure 21 (d) is based on multiple configurations of different markers attached to a vehicle. It includes five close distance recordings between 5 and 15m and six far distance recordings between 20 and 60m. It is used to directly compare three different marker configurations, which are discussed in Section 5. Figure 22 specifies the occupied marker area of all four configurations[5]. Second, the dataset of Figure 21 (c) is used to experimentally determine the detection range of different WhyCon-based markers, recorded in 5m steps up to 55m. Both subsets show the characteristic of low exposure beginning at a distance of 50m. This is caused by the shadow of a row of trees that darkens the part of the image with the pictured vehicle.
All real-world setups are also simulated, as exemplary shown in Figure 23, to facilitate a deeper evaluation of the applied marker configurations. The reference point RP is placed in the center of the middle AprilTag for datasets (a, b) and in the center of the large WhyCon marker for dataset (c).



(a) Type 1 (b) Type 2 (c) (d)

Figure 23: Exemplary simulated left camera subimages from the proposed datasets. (Increased contrast and brightness for better illustration)

[5]The marker dimensions are listed in Appendix B.3.

4.3 Simulation

In general, real-world benchmark datasets are rather difficult and time consuming to produce. In contrast, the creation of synthetic datasets is only limited to the available computational power and the number of implemented variable parameters. Furthermore, simulated data provides perfect noiseless ground truth data.
The goal of the proposed simulator is to complement the real-world datasets with more dissolute synthetic datasets, which are generated with varying properties. Next to the extensive evaluation and comparison of different RPV-methods, it also allows to "[...] investigate the influence of camera or scene properties on the [distance evaluation], to prototype, design, and test experiments before realizing them in the real-world, [...]" (Ley et al., 2016, p.4).
The evaluation of this work is based on a Monte-Carlo-Simulation, which is used to statistically estimate the uncertainty of the estimated distances and to analyze correlation to input parameters. For a number of trials, values are sampled from the assigned probability density function (PDF) of the individual parameter. For simplification, they are assumed to be independent in this work. The estimated distances of all iterations form a PDF that is used to define the resulting uncertainty. The implementation of the Monte-Carlo-Simulation is carried out according to the step-by-step procedure of (JCGM, 2008b)
Table 2 shows all parameters that are varied between each iteration. They are divided into two categories. First, in the simulation stage all parameters that variate the scene are changed for each iteration. And second, uncertainties of all calibration parameters are modeled in the application stage. In the following, the implementation of the Monte-Carlo-Simulation in the evaluation pipeline of Figure 19 is explained. The explanation starts with the embedding of the extended shader pipeline and continues with the realization of the application stage. Finally, the main evaluation procedure is introduced, which is used in the further course of this work.

Stage 1 - Aggregation	Stage 2 - Application
Variation • in image exposure • by camera pose trembling • of image noise (on/off)	Uncertainty of • interior camera calibration • stereo calibration • marker calibration

Table 2: Varied parameters in the evaluation pipeline for each iteration

4.3.1 Simulation Stage

The simulation stage is based on the extended rendering pipeline of Section 3.4.2. Figure 24 illustrates how this pipeline is embedded in the aggregation stage to realize and update all specific scene and iteration specifications.
First, the spatial correlations of all objects of the scene are represented in a hierarchical fashion, the scenegraph (Akenine-Möller et al., 2008, p.658). This graph represents a tree with objects as nodes and three-dimensional homography transformations as

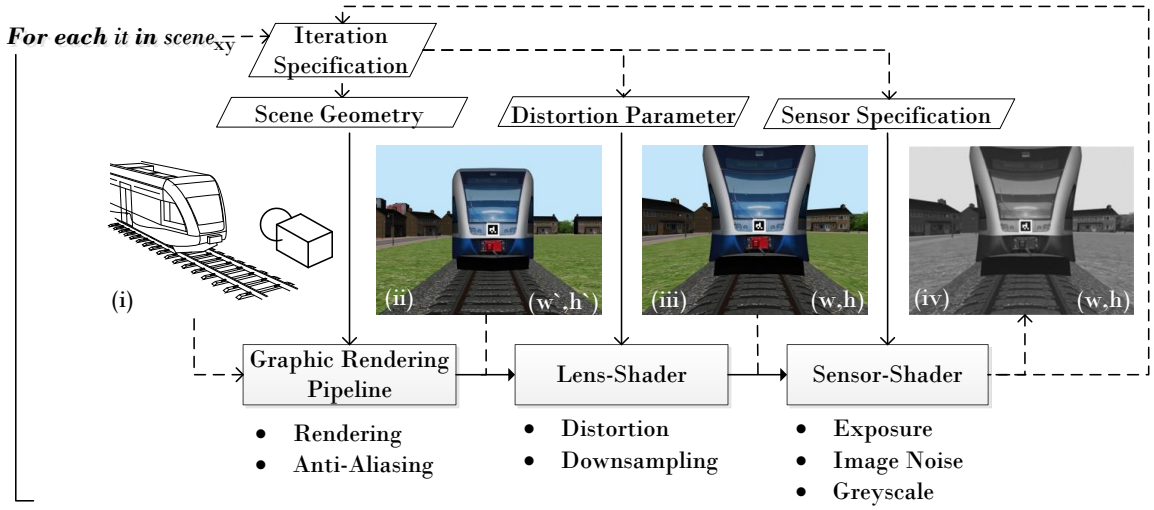


Figure 24: Embedding of the shader pipeline

edges. Figure 25 shows the important elements of the applied scenegraph. Starting in the world coordinate frame, it starts with the transformation H_{V2W} from the vehicle coordinate frame. In the example of Figure 18 (p.21), it describes the position of a train in world coordinates, visualized with a mesh of a train [Blend1] bound to this node. To this object, several markers T_n are attached. The asterisk of their homography marks that this transformation stays unchanged in the simulation, but will be noised in the application stage to model calibration uncertainty. On the same level, a node for the reference point RP is defined that represents the point on the vehicle to which the distance is to be estimated. To this node, the position of the left camera CL is set by the matrix H_{CL2RP}, defined by the to be evaluated distance d and view-angle α. To variate the position of the projected vehicle in the image, the orientation and position of the camera is minimal varied by the transformation $H_{Tremble}$ between each iteration[6].
Then the defined scene is rendered based on the camera model with the extended image borders (w',h'). This modification of the texture size includes on the one hand a summation of the offsets o_u and o_v to handle the complete distortion (see Section 3.1.2) and on the other hand it is scaled by the supersampling factor s, which is set to 4. The supersampling is directly embedded in the distortion step. As formulated in Equation 10a, each pixel (u,v) of the distorted image (iii) is mapped to the larger ideal image (ii) by the inverse distortion equation, the offsets o_u, o_v and the scaling s. Within the

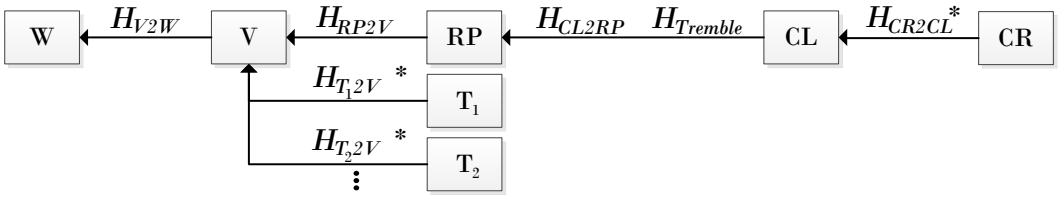


Figure 25: Important components of the scenegraph

[6]Details are provided in Appendix B.1.

ideal image the neighborhood (i_s,j_s) of s^2 pixels is sampled and averaged. The samples are interpolated by bilinear interpolation (Akenine-Möller et al., 2008, p.158). Besides supersampling, multisampling and mip-mapping are applied, but already implemented by [OSG].

$$p_{iii}\left(\begin{bmatrix}u\\v\end{bmatrix}\right) = \frac{1}{s^2}\sum_{i_s=0}^{s-1}\sum_{j_s=0}^{s-1} p_{ii}\left(\begin{bmatrix}u'\\v'\end{bmatrix} + \begin{bmatrix}i_s - \frac{s-1}{2}\\ j_s - \frac{s-1}{2}\end{bmatrix}\right) \tag{10a}$$

$$\text{with} \quad \begin{bmatrix}u'\\v'\end{bmatrix} = \left(\begin{bmatrix}o_u\\o_v\end{bmatrix} + \text{distort}^{-1}\left(\begin{bmatrix}u\\v\end{bmatrix}\right)\right) * s \tag{10b}$$

Finally exposure is applied, which scales all pixel values by a factor that is varied for each scene and image noise is added. Further lens and sensor effects such as blurring and vignetting are not applied due to incomplete information about these properties.

4.3.2 Application Stage

In the application stage, all RPV-methods are applied to each iteration of all scenes. As illustrated in Figure 26, before an iteration of one scene is processed, all geometric calibration parameters are resampled to employ false values with regard to their calibrated distribution.
In the simulation itself all calibration parameters are fixed to the values estimated from the calibration of the real-world camera [DLRStereo]. The calibration uncertainty is subsequently modeled in the application stage by sampling from the parameter distribution around the simulated value. In contrast to directly simulate the calibration uncertainty in the simulation stage, this results in the loss of different unknown image effects. However, this variant is used for several reasons. First, applying the interior camera calibration uncertainty in the simulation, thus recreating the distortion lookup table at each iteration makes it computationally infeasible to apply an extensive Monte-Carlo test[7]. Second, the calibration of the markers cannot be simulated since variation in their pose could lead to overlapping with the rigid modeled vehicle surface. And third, the uncertainty of the stereo calibration H_{CL2CR} (Figure 7 p.8) is also modeled in this stage to cleanly separate the application of variation and calibration uncertainty.
For each RPV-method, an object is initially created to avoid setting up all buffers

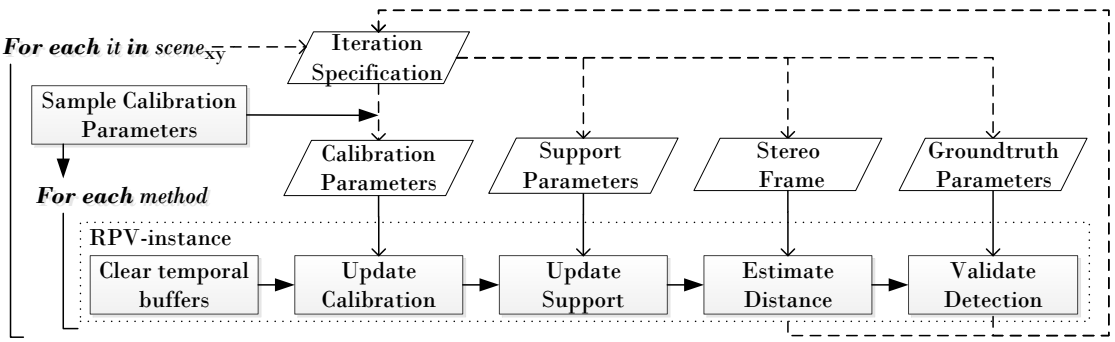


Figure 26: General procedure in the application stage

[7]The generated frame rate drops from around 15 frames per second to around 3 per minute.

for each iteration. However, these buffers are cleared for each iteration to facilitate independent estimations, as marked in Figure 26. Then the target parameters with meaning of the calibration parameters and additional support parameters are updated. Afterwards the distance is estimated for the current stereo frame. Finally, ground truth data is used to verify the detection of the required markers to exclude influences by erroneous detections during the statistically evaluation of the estimated distance.

4.4 Evaluation Procedure

To evaluate and compare the different RPV-methods, various evaluation criteria are applied to evaluate the accuracy and uncertainty of the estimations and to analyze correlations to specific input parameters.
Figure 27 illustrates how the different methods are compared to each other. The columns show two different simulated experiments. They differ in the way that (i) only applies variation *(+Var)* and (ii) modeles variation and calibration uncertainty *(+Var,Unc)* (see Table 2, p.25). Both are investigated for two primary parameters. First, the rows of the plot cluster correspond to the setting of the view-angle with 0° and 30°. Second, the entries on the x-axis of each plot correspond to the distance with the values 5.88m and 24.2m. For visualization, they are rounded while their exact values can be found in Appendix B.3. For each distance, a set of images is simulated based on the number of iterations. This number is specified by the pictured variable *fpb* (frames per box). Two methods are applied on the same images of the current distance. This results in a box plot (Tukey, 1977) for each combination of a method, distance, view-angle and experiment. The box plot shows the Quartiles with a box and a middle line for the median. The value range is shown by the whiskers, which are set to the percentiles 2.5 and 97.5 to exclude outliers. This plot cluster can be used for different primary parameters, experiments, methods and varied parameters during each iteration. The order of primary parameter B and the different experiments can be switched. If the estimations of one box plot has a detection rate of less than 80% it is not shown. *CL* states that the specific estimation is based on the left camera.

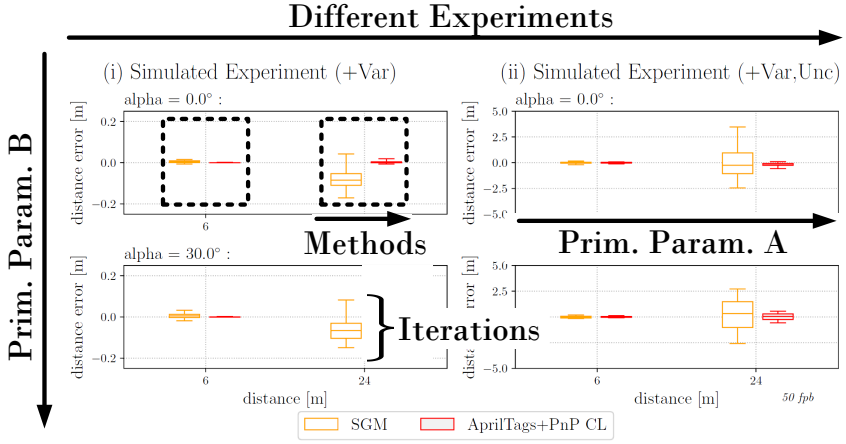


Figure 27: Explanation of the evaluation procedure and plots

5 Integration of Methods

This chapter describes the RPV-methods applied in this work. First, the usage of the fiducial markers AprilTag and WhyCon is outlined. Therefore, different configurations are proposed for each marker, which are verified in a preliminary evaluation to continue with the individual best configurations in the final evaluation phase. Finally, it is explained how SGM is applied and how the markers could be used in a stereo setup.

5.1 Integration of AprilTags

In this section, different configurations of AprilTags are tested and evaluated based on the experiment setup of Figure 21 (a) and (b). Subsequently, this knowledge is used to propose an AprilTag configuration, which is used for the comprehensive evaluation.

5.1.1 Application

When using AprilTags, the distance is estimated by applying a PnP method on the corners of the detected AprilTags (see Section 3.3.1). The advantage is that multiple AprilTags can be applied at the same time. Formulated in Algorithm 2, multiple markers M_{calib} can be attached to the same vehicle, each defined by its pose H^*_{T2V} in vehicle coordinates and the id of the marker. (Square brackets represent lists.)
The algorithm starts by extracting the AprilTags from the image, while m_T holds the respective four corners. Based on the id, the extracted AprilTags are assigned to the defined markers of M_{calib}. If more than one extracted AprilTags have the same id, multiple candidates are created. If one is missing, only the detected AprilTags are used. For each candidate, all image points x_{image} and corresponding world points x_{world} are stored in M_{corr}. For each candidate C_t, the pose of the camera to the vehicle H_{CL2V} is estimated by RPnP and by using the average reprojection error r_t, the best candidate

Algorithm 2: Application of AprilTag as RPV-method

Data: Image - current frame, $M_{calib}[H*_{T2V}, id]$ - list of markers
Result: d - distance to left camera, b - success of the estimation

$M_{extracted}$ $[m_T, id] \leftarrow$ extractMarkers(Image) ▷ Generate Correspondences
$M_{assigned}$ $[[m_T, H_{T2V}*]] \leftarrow$ assignMarkers($M_{extracted}, M_{calib}$)
M_{corr} $[[x_{image}, x_{world}]] \leftarrow$ getCorrespondences($M_{asigned}$)
$H_{CL2V}, r_b, C_b, b \leftarrow I, \infty, [], False$ ▷ Find best candidate
for ***each*** C_t *in* M_{corr} **do**
  $H_t \leftarrow$ linearResection(C_t)
  $r_t \leftarrow$ calcReprojError(C_t, H_t)
  if $r_t < r_b$ ***and*** $r_t < r_{thresh}$ **then**
    $H_{CL2V}, r_b, C_b, b \leftarrow H_t, r_t, C_t, True$
$H'_{CL2V} \leftarrow$ iterativeResection(C_b, H_{CL2V}) ▷ Refine best candidate
$d \leftarrow$ norm(getTranslation($H'_{CL2V} * H^{-1}_{RP2V}$))

C_b is selected. r_{tresh} represents a maximum allowed error (50px) to sort out obviously wrong estimations. Finally, the pose is refined by an iterative PnP approach, before the distance d from the vehicles reference point to the left camera is estimated.

5.1.2 Preliminary Evaluation and Summary

The preliminary evaluation of Figure 28 compares the usage of two different AprilTag types. It is based on the setups (a) and (b) of Figure 21, applied in real-world and simulation. Type 1 applies three small markers with high code density and Type 2 shows large markers with smaller code density. Additionally, the applications of all three markers (AprilTags #3) and only the middle one (AprilTags #1) are investigated.
The real-world experiment (ii) shows an increased deviation to the ground truth distance when using only one marker. This is caused by a strong overexposure of the recordings that makes the marker appear smaller, illustrated in Appendix B.3. This effect is also present in the simulated experiment (i), revealed by the long upper whisker when using only one marker (blue) of Type 2. This effect can be bypassed by using multiple markers (red) since the influence of inaccuracies of the corner detection is balanced by a greater number of correspondences for PnP (see Section 3.2.2). When comparing the different marker types, it is striking that Type 2 has a greater detection range. Furthermore, the number of used markers also increases the detection range, because only one marker needs to be detected for a successful distance estimation. But also because using only one marker leads to a more frequent failure of RPnP itself.

To summarize, this experiment shows the superiority of large markers with small code density. Also, using three marker increases the accuracy of the estimation due to more correspondences. However, to provide a total marker area that is comparable to other RPV-methods, only one large and two small markers are applied in the final experiment of Figure 28 (a). This setup should support a wide detection range due to the large marker and an accurate estimation for smaller distances due to three usable markers.

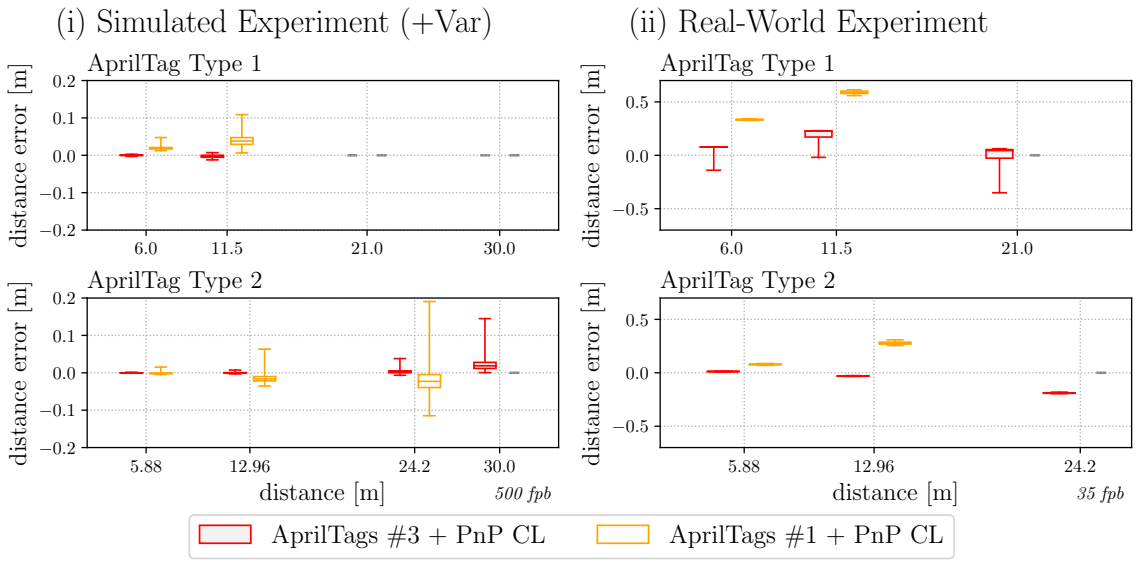
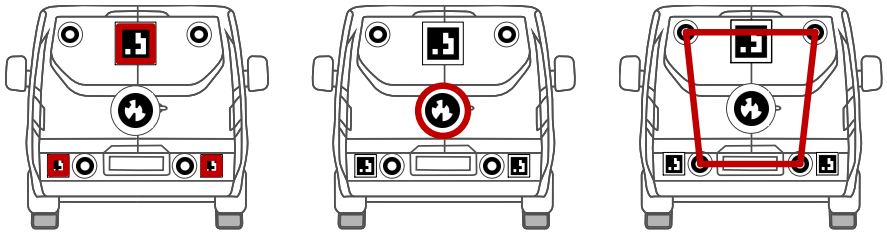


Figure 28: Preliminary evaluation of AprilTags



(a) AprilTags+PnP (b) WhyCon+Circle (c) WhyCons+PnP

Figure 29: Specification of marker configurations for the evaluation

5.2 Integration of WhyCon

This section explains the application of WhyCon in the RPV-setup to estimate the distance. That includes an adaption of the thresholding method to ensure a reliable detection of the pattern in environments with varying lightning. Also, a coding system and extraction is proposed. Its purpose is to validate the detection range in comparison with a pattern without code and not to compete with similar existing methods such as WhyCode (see Figure 6, p. 6). A corresponding preliminary experiment is presented.

5.2.1 Application

For a reliable detection of WhyCon patterns in an environment with varying lightning, the thresholding variant of WhyCon using the threshold estimated in the previous frame is not sufficient. It would decrease the detection rate and accuracy since the selection of the threshold directly influences the detection of the pattern borders. Therefore, a pre-detection step is applied in this work, as shown in Figure 30. For each processed frame, an image pyramid is constructed and each level is submitted to an adaptive mean thresholding [OpenCV] with a small neighborhood area. The WhyCon detection is applied on all resolution levels to ensure the detection of patterns of any size regardless of the chosen neighborhood size. The results of all levels are collected while only one entry is hold for each double detected WhyCon pattern in different levels. Then for each detected pattern m'_T, its black and white pixel values are sampled in the original image. This is done by taking each 50 samples of the corresponding area and using the

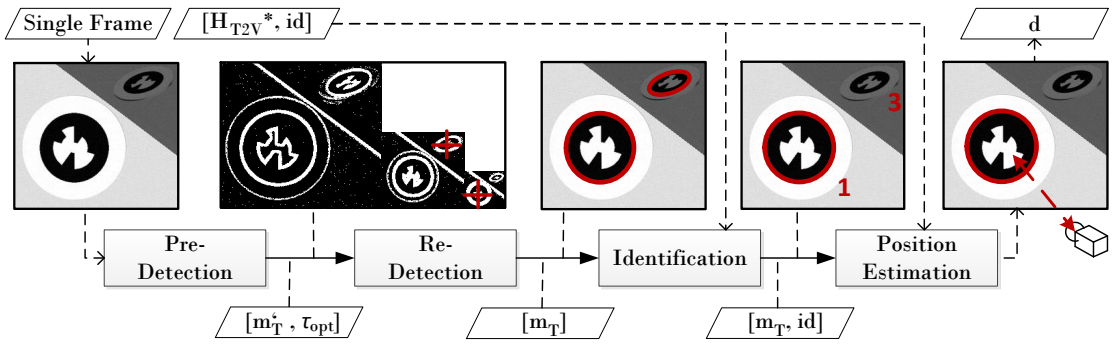
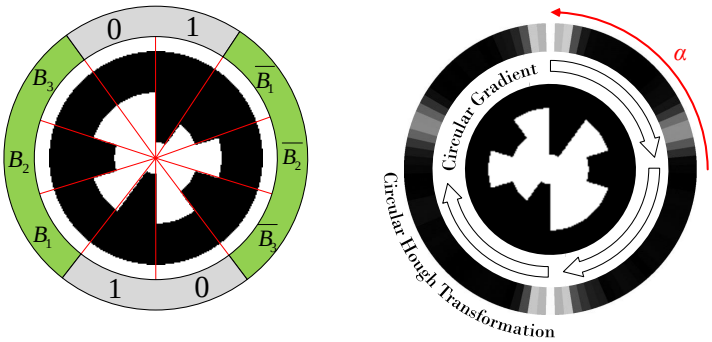


Figure 30: Applied processing chain with WhyCon

medians to calculate the optimal threshold τ_{opt} for each individual pattern. Based on this threshold, each WhyCon marker m_T is re-detected with optimal thresholding. As discussed in Appendix A.3, the speed advantage of WhyCon by tracking the pattern is lost, but the proposed extension is robust to variation in exposure.
In the third step, the patterns are identified that most likely represent the attached WhyCon markers on the vehicle. It is distinguished between two setups shown in Figure 29 (b) and (c).
First, in Figure 29 (a) a single WhyCon marker with the proposed coding of the next section is attached to the vehicle. The purpose of this configuration is to evaluate the performance of the distance estimation based on the circle of one large pattern. Due to the attached coding, the pattern can be directly identified by its code. In it the comprehensive evaluation the id is not used, as described in Section 5.2.4. Instead, it is identified by its projected size with the assumption that all faulty detected WhyCon pattern are smaller than this projected pattern or have a less circular shape. This assumption holds true for all considered datasets in this work. Moreover, due to the applied code the inner circle does not represent a circle anymore, which is why the proposed correction of the circle semiaxis of (Krajník et al., 2014, p. 8) is not applied.
Second, Figure 29 (c) shows a configuration with four normal WhyCon markers whose detected center points are used for PnP while using a method similar to Algorithm 2 (p.29). The idea is to reduce inaccuracies by employing a large quadrangle spanned by four small patterns. They are identified by their same appearances and spatial correlations during the creation of candidates in the method *assignMarkers*. Thus, four pattern form a candidate, if the following condition regarding their size is full filled. The square brackets include the four patterns and $\boldsymbol{e_0}$ states the first eigenvector of the ellipse of m_t (see Section 3.2.3).

$$\frac{max([m_T.\boldsymbol{e_0}])}{min([m_T.\boldsymbol{e_0}])} < 1.5 \tag{11}$$

The patterns are then assigned to the attached WhyCon markers based on their spatial correlation. Exemplary, the projected left-down pattern of Figure 29 (c) has a smaller x-value then the two patterns on the right and a larger y-value than the two above. Only one possibility remains for each foursome combination.



(a) Coding Definition (b) α-Shift Estimation

Figure 31: Visualization of steps to estimate the angular shift of the code

5.2.2 Proposed Code System and Extraction

Inspired by WhyCode (Lightbody et al., 2017), a coding system and code extraction is proposed in this section. The goal of this implementation is to evaluate the detection range of WhyCon with code identification and not to compete with WhyCode itself. The proposed coding system is presented in Figure 31 (a). The code is attached to the inner border of the black circle. Two large opposing excesses define the beginning of the hidden code. Three bits on the left that are used to store the id are negated and mirrored on the right side.

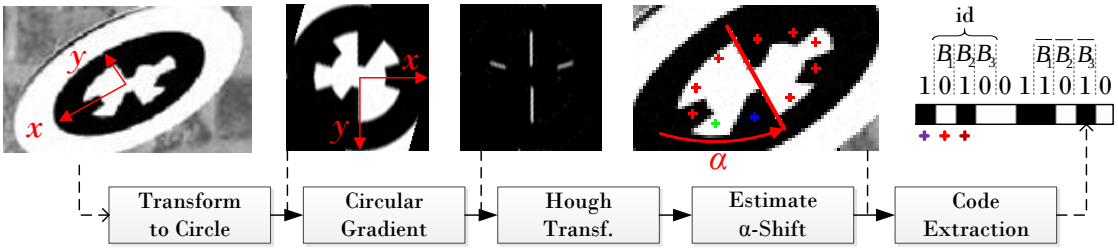


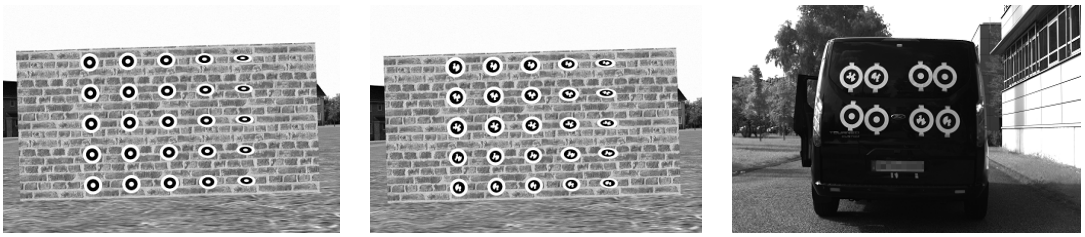
Figure 32: Pipeline for extracting the binary code

The code extraction algorithm is summarized in Figure 31 (b) and detailed in Figure 32. First, the subimage of the detected ellipse is transformed into circle coordinates with a fixed resolution of 31^2px. This transformation ensures the correct weighting of the gradients in the Hough space. Second, a clockwise circular gradient is applied by calculating the image gradients in x and y direction with the sobel operator (Jähne, 2005, p. 365) first and projecting them in circular direction afterwards. The circular gradients are then registered in an one-dimensional Hough space to find the angle that corresponds to the beginning of the code. Since two possibilities remain, the assumption that the pattern can not be upside down in the image is employed. With the α-shift by hand, the n pixel values p_i of each bit are sampled and a binary value b_i is assigned based on a spatially-varying threshold formulated in Equation 12. This assignment method shows the necessity of equal numbers of black and white bits.

$$b_i = \begin{cases} 0 \text{ if } p_i < \frac{1}{n}\sum_{j=1}^{n} p_j \\ 1 \text{ else} \end{cases} \tag{12}$$

5.2.3 Preliminary Evaluation - Coding

This preliminary evaluation investigates the detection range of WhyCon marker with and without an attached binary code. Figure 33 shows the used datasets. The simulated setup (a) includes multiple WhyCon in front of a wall with different tilts. This setup is used to estimate the detection range for scenes with variation (WC Detection). (b) shows multiple WhyCon extended with a binary code. This setup is used to apply the WhyCon pattern detection either with (WC* Identification) and without identification (WC* Detection). Finally, on the real-world setup (c) all three methods are applied to confirm the results from the simulated experiment. All marker of these setups have a total diameter of 24cm, including the outer white circle (see Appendix B.2).



(a) Simulated WC (b) Simulated WC* (c) Real-World

Figure 33: Cropped Extracts from the experiments of WhyCon detection

One property of the simulated experiment is that all detection rates start decreasing quite early at around 30m, which is caused by the strongly angled markers. The displayed line at a detection rate of 0.5 seems to be suited for a comparison of the methods since it still includes all slightly tilted markers. The experiment (a) of Figure 34 shows that the extended WhyCon pattern has an even better operation than the normal WhyCon. This is caused by the adaptive thresholding, which favours a different black ring width than the original method as deepened in App. A.3.2. When applying the identification of the code, the detection range drops by 10m at a detection rate of 0.5. Both observations can be retrieved in the real-world experiment that only contains slightly tilted markers. First, the detection range of both markers types is nearly equal. And Second, the identification reduces the detection range of the extended WhyCon marker by 10-15m. Thus, the code of this marker size was always successfully identified up to 25 meters, while the detection of the extended marker goes up to 40m with a 100% detection rate. This observation implies a loss of the detection range of 30%, when applying code extraction.

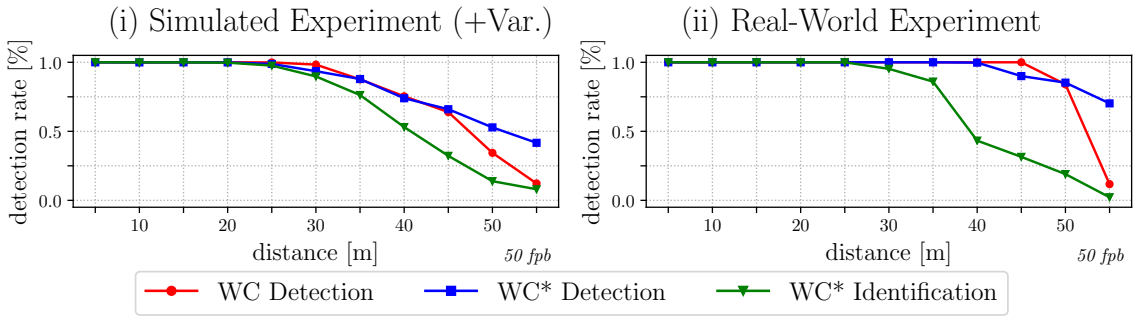


Figure 34: Analysis of the detection range for different WhyCon patterns

5.2.4 Summary

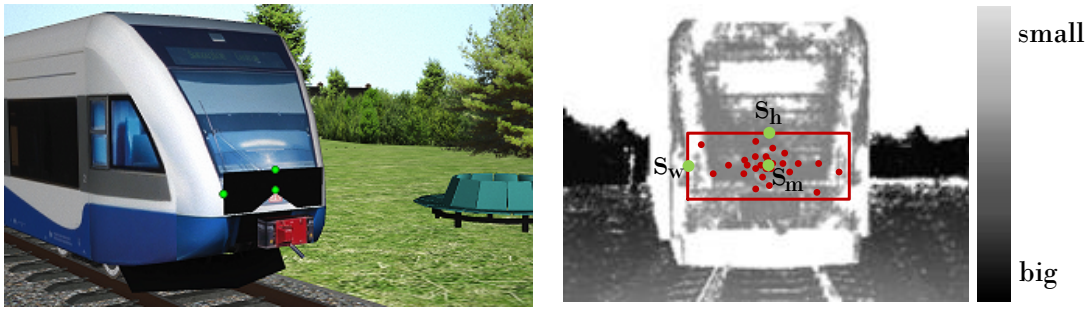
Two WhyCon configurations are chosen for the comprehensive evaluation. First, a single large WhyCon marker with attached binary code. However, the code identification is not applied further since the advantage of the extended WhyCon marker is that it can be detected even though the code can not be extracted. Second, four single WhyCon markers without attached code are used to apply the PnP approach.

5.3 Integration of Stereo Methods

In this section, the application of different stereo methods for RPV is explained. This includes an approach that explores the potential of SGM. In addition, the detected markers in both camera images are used for simple triangulation.

5.3.1 Application of SGM

The result of Semi-Global Matching is a disparity map, which allows to estimate the depth of each pixel by triangulation. From this disparity map all disparities that belong to the preceding car need to be classified, which requires a labeling of the vehicle in the image. This could be done by using a vehicle detection and classification method (Sivaraman and Trivedi, 2013a). However, this preliminary step makes the estimated result dependent on the quality of the classification. This is to be prevented in this work since the focus is to test the potential of estimating the distance by SGM. Thus, the vehicle is labeled in the image by the support parameters s_m,s_w,s_h. During simulation, these points are defined on the three-dimensional vehicle mesh, as illustrated in Figure 35 and are then projected into the camera. For the real-world experiment, they are labeled by hand for each different position.
The image points s_m,s_w,s_h are then used to create a rectangle, as shown in Figure 35. For each pixel of this subimage, a sampling weight is assigned based on a two-dimensional Gaussian distribution with the mean s_m and sigmas $\sigma_x = \frac{1}{3}|s_{m_x} - s_{w_x}|$, $\sigma_y = \frac{1}{3}|s_{m_y} - s_{h_y}|$. After normalization, this distribution is used to sample 101 disparities by stochastic universal sampling. For all samples, the depths of the pixels are triangulated. Finally, from all distances the median is chosen as the representing vehicle distance. This sampling is implemented to compensate single outliers, which can occur due to image noise and light reflections. Though, if the rectangle contains a bulge of the vehicle surface, a constant bias can occur in the statistical evaluation, which is not considered in this implementation.



(a) Reference points (b) Sampling in disparity space

Figure 35: Illustration of the SGM application

5.3.2 Triangulation of Markers

If markers are applied to the vehicle and the scene is observed by a stereo-camera, then it is convenient to apply triangulation on the detected pattern in both images to produce another potentially independent measurement. Therefore, either one reference pattern of AprilTag and of WhyCon is used for triangulation.
In the case of AprilTag, the pattern with the correct id is matched between both images. All four points are triangulated to estimate their three-dimensional position to the left camera. The distance is then calculated by the distance to the center of the marker, defined by the mean of the corner positions. In the case of WhyCon, a large pattern attached to the vehicle is used for triangulation. The pattern is identified as explained in Section 5.2.1.
In the case that the reference point is not the center of the considered marker, the distance is corrected by the Pythagorean theorem with the assumption that the reference point is on a vertical line to the estimate marker center. Thus, this correction is not generally applicable, but it is sufficient for the forthcoming consideration and only used to correct the distance of the estimation with the large AprilTag of Figure 21 (c).

5.3.3 Preliminary Evaluation and Summary

Figure 36 shows a short comparison of the SGM and the marker-based triangulation methods. The configuration of Figure 21 (c) is simulated for two small distances (i) and two far distances (ii). The experiment shows that the triangulated distance by the AprilTag is most accurate for short distances, but not applicable for far distances since the pattern is not detected anymore. WhyCon is more accurate than SGM for short distances. This is because the triangulation of WhyCon is based on the center of the circular pattern, which can be determined with high precision. This advantage decreases for far distances, because the number of pixels used for the center determination gets smaller. The consequence is that WhyCon shows a larger spread for far distances than SGM.
To reduce the number of compared methods in the following evaluation, only SGM is considered at first as representative for the stereo methods. The triangulation of the markers is pick up again in Section 6.3.

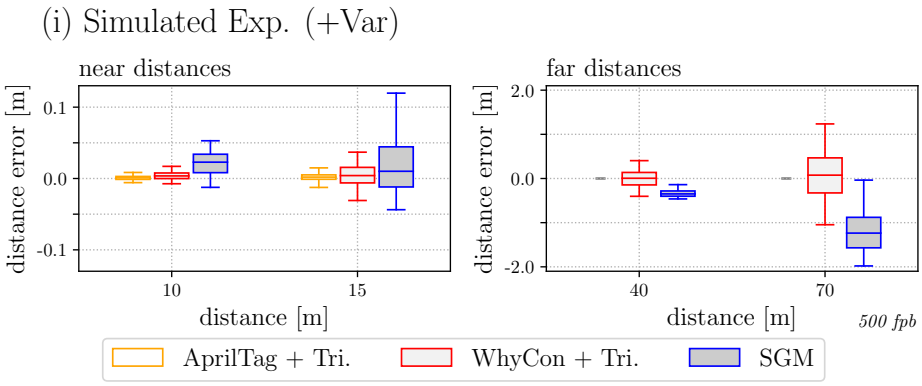


Figure 36: Comparison of SGM with marker-based triangulation

6 Evaluation

In this section, the selected configurations of Figure 29 (p.31) and the SGM-based approach are evaluated, with respect to the formulated research questions. For an explanation of the plot characteristics, please revisit section 4.4 (p.28).
Concerning Research Question (1), the general potential is investigated by concentrating on specific variation parameters in simulation. That includes the distance, the view-angle and image exposure. During these simulated experiments, the calibration uncertainty is not modeled to retain noiseless results. Then the influence of calibration uncertainty on the different methods is investigated, motivated by Research Question (2). Based on a Monte-Carlo-Simulation (MSC), the uncertainty of the methods are estimated and compared. Then, the correlation of the methods with specific calibration parameters is exploited and the discovered correlations are analyzed in more detail. Finally, it is investigated how the methods could be combined to yield a more accurate, robust and less uncertain estimation to answer Research Question (3).
A few of the experiments refer to the appendix, which contains tables with specified information. The computational time is briefly considered in Appendix A.4.

6.1 Qualitative Comparison

In this section, the methods are evaluated with focus on their accuracy when varying the distance and the view-angle to the preceding vehicle as well as the image exposure.

6.1.1 Distance

Figure 37 shows a comparison of the RPV-methods pointed out in the previous section based on the distance to the preceding vehicle. Therefore, a simulated and a real-world experiment are conducted based on the setup of Figure 21 (c, p.23). For both experiments, three short and three far distances are evaluated in Figure 37. The simulated experiment (i, left) shows a high precision of the PnP-based methods for short distances. Although, the spread of AprilTags+PnP is suddenly increasing at a distance of 20m. This is caused by the missing detection of the smaller AprilTags and the accompanying dependency on image exposure, discussed in Section 6.1.3. Related, the estimation based on the single WhyCon shows a relatively large spread for short distances caused by the same dependency on image exposure. However, Figure 37 (i, right) shows the advantage of WhyCon+Circle manifested in its large detection range, even though this marker configuration has the smallest occupied area of all three considered marker configurations of Figure 22 (p.24). Finally, the SGM approach provides good results comparable to WhyCon+Circle. Its results for the large distance indicate a smaller correlation to the varied parameters than WhyCon+Circle, but show an increasing bias of the average distance deviation for larger distances, which reveals the limitations of the disparity sub pixel resolution of this matching algorithm.
The results of the real-world experiment of Figure 37 (ii) confirm these observations. Please note that the real-world experiment only represents a snapshot of all applied variations since the 50 frames at each distance are taken with the same pose, which

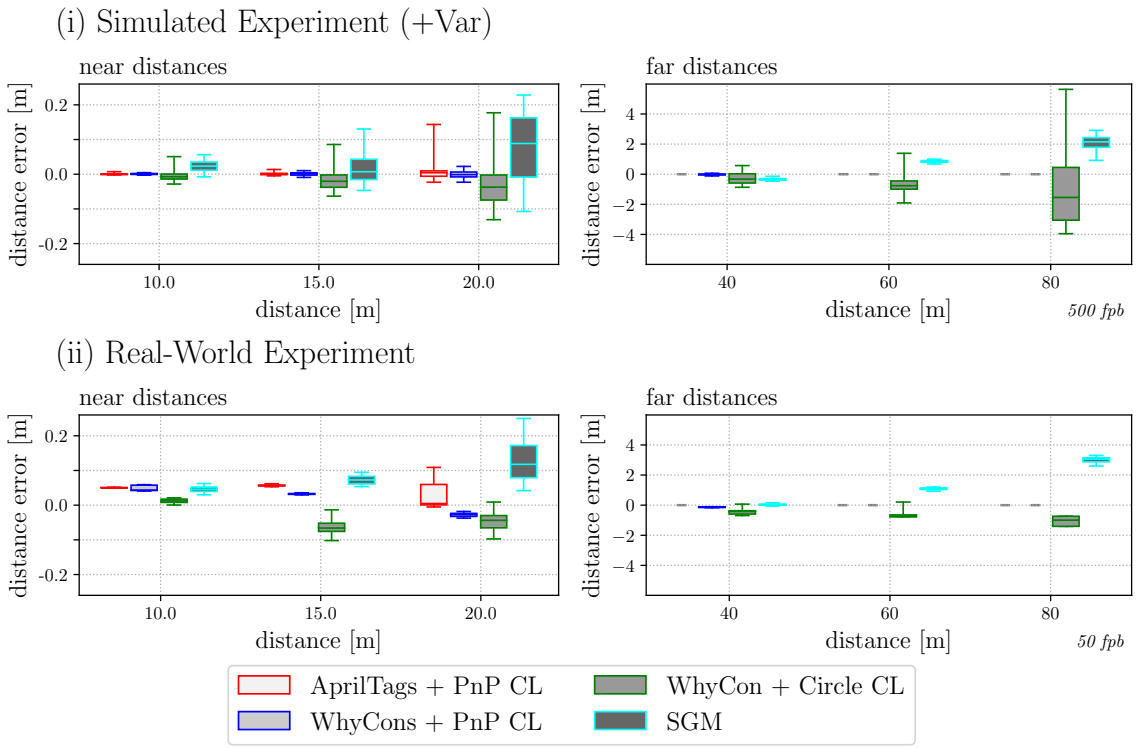


Figure 37: Comparison based on simulation and a real-world experiment

makes image noise the only varied parameter. Also, the calibration parameters represent only a fixed but random selection based on the individual distribution. The calibration uncertainty is considered for the simulation in the Section 6.2. The real-world experiment (ii, left) shows an average bias for all methods of around 4cm, which could be caused by inaccurate recordings of ground truth values. And similar to (i, right), (ii, right) indicates a wide application range of WhyCons+PnP, which is considered in the next section in more detail. This method shows a high spreading in the simulated data at a distance of 80m, which is caused by the variation of the scene. Especially the image exposure has a great influence, as discussed in Section 6.1.3 (p.41). This large spreading is not present in the real-world experiment (ii, right), where the frames differ only in image noise at each distance. The influence of image noise on WhyCon is briefly discussed in Appendix A.3.3.

6.1.2 Application Range and View-Angle

Figure 38 shows an evaluation of the application range (i) and the accuracy for two selected distances (ii) for different view-angles, accomplished by varying the parameter α of Section 4.1 (p.21). The consideration of the view-angle is necessary, because trains have different flat noses and the view-angle changes in curves. Figure 39 illustrates the considered α values.
The bars of Figure 38 (i) illustrate the application range for each method with a marking of the success rate of the estimations for 95%, 80% and 50%. For this experiment, 100 frames are simulated and evaluated at each noted distance for each angle. Based on

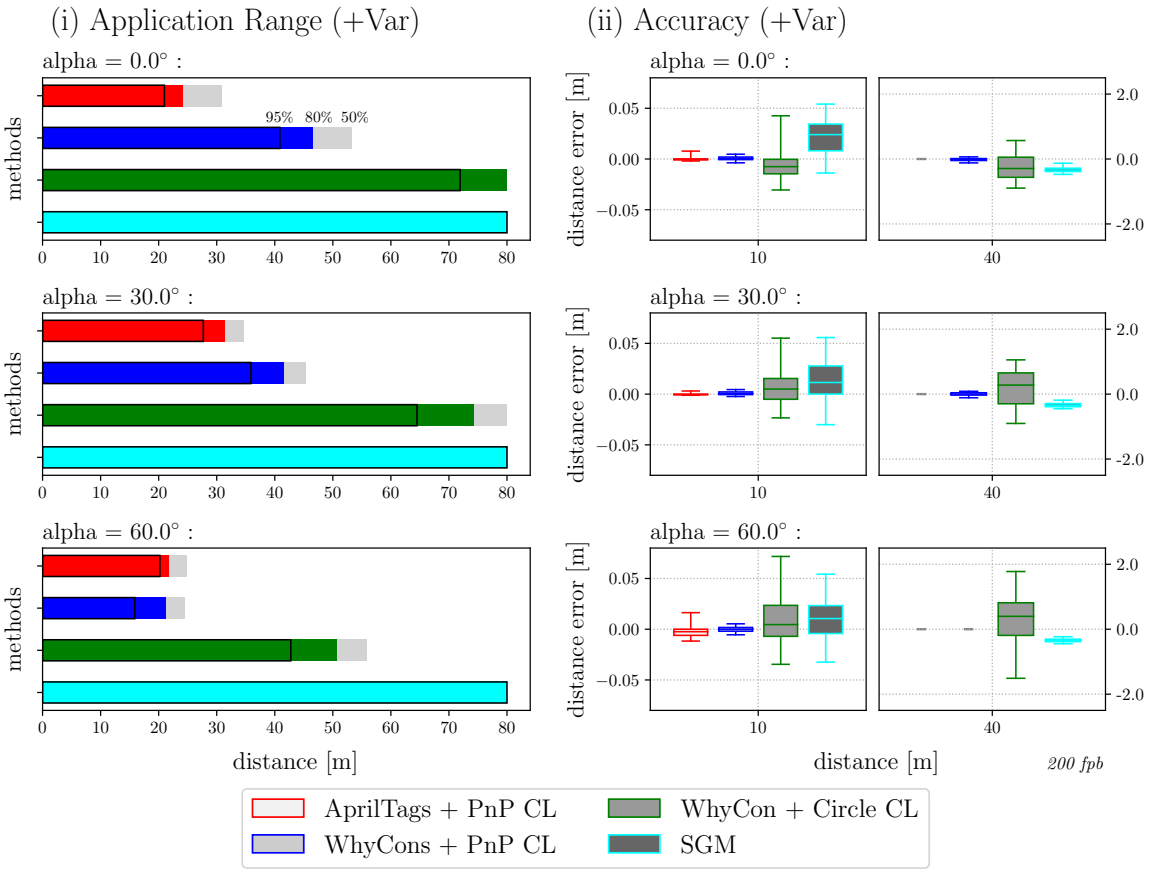


Figure 38: Evaluation of the application range for different view-angles (App. B.4.1)

this estimation, the success rate is estimated and interpolated between the evaluated distances to provide a good impression of the method behaviors. It is obvious that the application range of the marker-based methods decrease with a more acute view-angle. However, unexpectedly is the higher application range of AprilTags+PnP at 30° than at 0°. At a distance of 30m and angle of 0°, the big AprilTag is still detected in almost all frames. But the subsequent distance estimation with RPnP based on the four corners of the marker frequently fails in this situation where the AprilTag is almost perpendicular in front of the vehicle. The application range of WhyCons+PnP is slightly worse than AprilTags+PnP at a view-angle of 30°. This is caused by the

(a) $\alpha = 0°$ (b) $\alpha = 30°$ (c) $\alpha = 60°$

Figure 39: Exemplary simulated images for the application range comparison. (Increased contrast and brightness for better illustration)

slight curvature of the vehicle rear, noticeable in Figure 39 (c). This leads to a rapidly occurring non-detection of the upper left WhyCon and since all four corners need to be detected for the PnP estimation, the application range is restricted. On the other hand, WhyCon+Circle provides a robust application range with a success rate of 95% up to 70m at a view-angle of 0°. But also drops down to 40m when greatly increasing the view-angle. The application of SGM is less meaningful at this point since the car is labeled in the image for this approach.
When considering the accuracy for the selected distances in Figure 38 (i), it is striking that the influence of the view-angle is negligible. An exception is WhyCon+Circle that shows an increasing inaccuracy for more acute angles, well visible at the distance of 40m in Figure 38 (i, right). This is caused by the increased view-angle that reduces the number of pixels inhabited by the projected marker, which has a similar effect as increasing the distance of the marker to the camera.

6.1.3 Image Exposure

When using fiducial markers for distance estimation, image exposure can have a relevant influence on the estimated distance. (Mangelson et al., 2016) has shown in a real-world experiment that the corner detection of AprilTags is highly affected by image blooming. It describes the blooming effect of white areas onto surrounding pixels, which varies for different exposure factors. They solved this problem by surrounding the AprilTag with small circles, whose center estimation is more robust to blooming effects. Thus, this effect needs to be considered for the application of the proposed marker configurations. Therefore, the different RPV-methods are examined for different exposure factors in simulation. Please note that this investigation is restricted in its universality since the applied simulator does not especially model image blooming and no suitable real-world experiment was conducted.

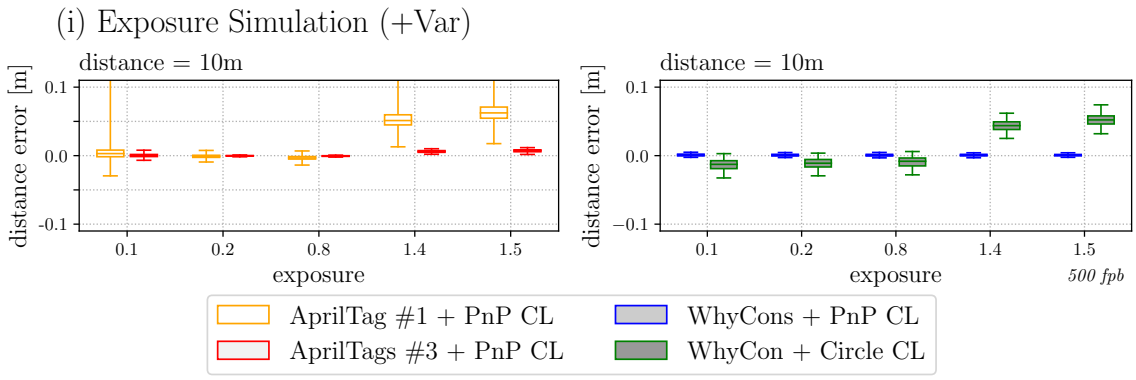


Figure 40: Consideration of image exposure (App. B.4.2)

Figure 40 shows the behavior of the approaches for variation of image exposure. Since Section 6.1.1 implied different behaviors for the application of a single (AprilTag#1) and multiple AprilTags (AprilTags#3), this experiment considers both variants. Also both WhyCon approaches are evaluated. SGM has shown that it does not significantly correlate with image exposure, which is why it is not considered in this section.

In the case of image exposure with a value of 1.0, the projected pixels of the markers black area have a grey value of 5 and the pixels of white areas a value of 255 in simulation. This implies that the exposure of 1.0 marks the transition to saturation. The single AprilTag shows a bias in the distance error for over exposure (>1.0). This is caused by the saturation that pushes the AprilTag borders inwards as described in (Mangelson et al., 2016). Figure 41 illustrates this effect. It shows the transition of the markers black and white area, which is quantized to pixels x_j and grey values for different exposure factors. When increasing the exposure from 1.0 to 1.5, the grey value of the border pixel x_i increases, the estimated border is pushed towards the black area, which makes the marker appear smaller in the image and thus, a too far distance is estimated in average. This effect does not occur for low exposure, because the relation between the values of the pixels $x_{i-1,}, x_i, x_{i+1}$ remains the same. This effect is also balanced when using multiple AprilTags since the relative distance of the markers to each other remains unchained. However, since the two AprilTags are rather small in the chosen configuration of Figure 29 (a, p.31), this effect occurs for larger distance, as observed in Section 6.1.1. This results in a long tail of the upper whiskers.

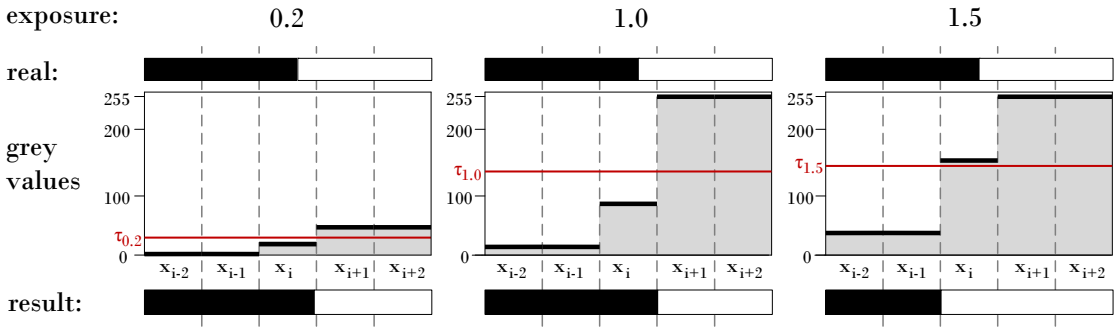


Figure 41: Illustration of the influence of exposure

The same effect caused by over exposure occurs by the application of WhyCon+Circle. Please note that the compensation of incorrect diameter estimation of (Krajník et al., 2014, p. 8) is not applied, as stated in Section 5.2.1. For this approach, Figure 41 also provides an illustration of the estimated threshold (red line) and the resulting binary assignment for each pixel. In contrast to AprilTags+PnP, WhyCon+Circle shows a slightly negative bias for the remaining exposure factor, which implies that the line detection of the AprilTags line detection is potentially more accurate than the hard assignment of WhyCons circle detection. Similar to the usage of multiple Apriltags, the estimation of PnP based on the four centers of four WhyCons is invariant to image exposure since the projected pattern is equally affected in all directions and the estimated center remains the same. Shown in (Mangelson et al., 2016) for circular markers by a real-world experiment.

6.2 Consideration of Calibration Uncertainty

This model concentrates on the addition of geometrical calibration uncertainty. The uncertainty of each method is estimated by a MCS. Afterwards, an extensive MCS is used to determine correlations of the methods to specific calibration parameters, which are then considered in more detail.
In this work, the term uncertainty refers to the standard uncertainty that is defined by the standard deviation of the PDF (JCGM, 2008a, p.3). All PDFs are assumed to be Gaussian. By the definition of the whiskers of Section 4.4, they show the 95% coverage interval (JCGM, 2008b, p.9) of the estimations in this section. The used calibrated parameters and their uncertainties are listed and substantiated in Appendix B.2.

6.2.1 Direct Comparison

Figure 42 repeats the experiment of Figure 37 (p.38, i) and additionally applies uncertainty of calibration parameters according to section 4.3.2. This includes uncertainty of the stereo calibration, interior camera parameter and uncertainty of the marker attachment on the vehicle, noted in Table 2 (p. 25). When comparing Figure 42 (iii) with Figure 37 (i), it is striking that the modeled calibration uncertainties increase the spreading and thus the uncertainty of all RPV-methods. The uncertainty of each method is represented by the standard deviation of the assumed out-coming Gaussian distribution, marked with three small marks. The middle mark represents the bias.

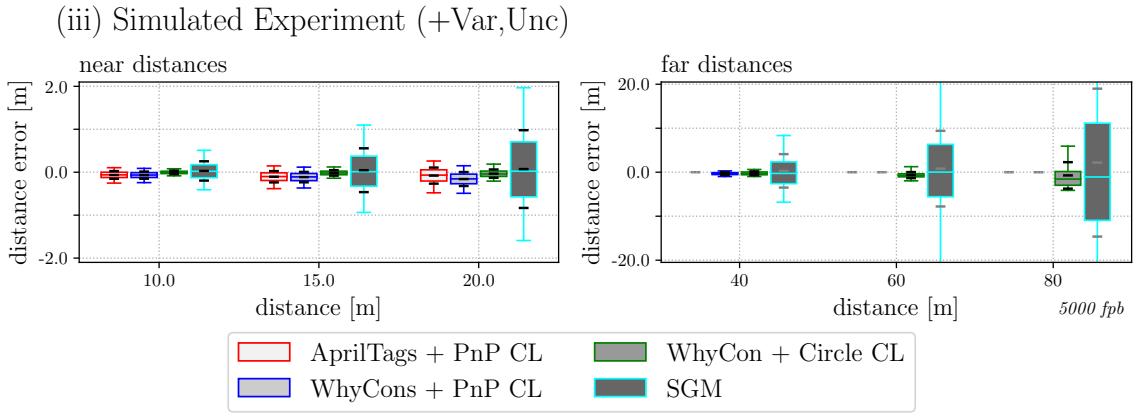
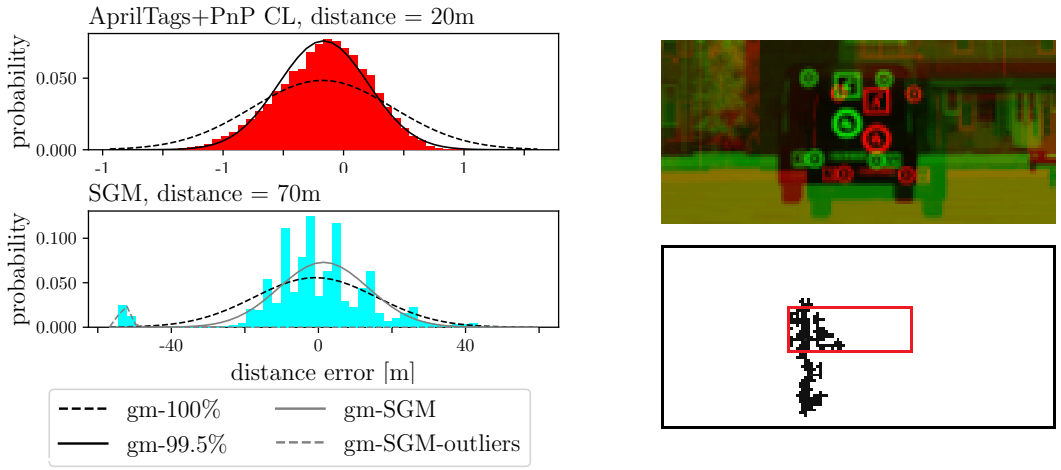


Figure 42: Comparison based on simulation with variation and uncertainty

When considering the marker-based methods for near distances, WhyCon+Circle has the smallest uncertainty. This is caused by a small dependency on all calibration parameters. The dependencies are explored in the next section. The two PnP-based methods show approximately the same behavior with a relatively large uncertainty compared to WhyCon+Circle. This is caused by the strong dependency on the marker calibration. In contrast, SGM shows a comparatively large uncertainty that is rapidly increasing with larger distances. This is caused by strong dependencies to many calibration parameters of the camera. The course of the pictured standard deviation shows a square rise of the uncertainty of the estimated distance to the examined distance. This matches the theoretical consideration of Section 3.2.1 with Figure 12 (p. 12).



(a) Examples with extreme outliers (b) Anaglyph and disparity image

Figure 43: Consideration of faulty outliers

The pictured standard deviations of Figure 42 exclude extreme outliers from the calculation. Figure 43 (a) shows two distributions that contain extreme outliers (outliers are not pictured). Concerning AprilTags+PnP, gm-100% represents the resulting Gaussian PDF that results if extreme outliers are included. The actual distribution is not well represent. Because of that, only the middle 99.5% of the sorted data is considered for all standard deviation estimations, resulting in gm-99.5%.
Second, SGM shows another characteristic for far distances in (a, bottom). The plot for 70m shows a concentration of outliers at a distance error of around -65m. Figure 43 (b, top) shows an example of a rectified image pair used for the SGM approach at 80m that shows the reason for this characteristic. It is rectified by a camera model composed of sampled calibration parameters. In such an extreme case, SGM is no longer able to match both images correctly since it only matches on the same image line. The resulting disparity map of (b, bottom) shows a sparse disparity estimation with only wrong estimations due to incorrect matching. To filter out these faulty estimations, the estimated results of all iterations of one scene with a distance less than $\frac{1}{3}$ of the current investigated distance are sorted out, if this condition concerns at least 0.5% of the already cropped data. Figure 42 (a, bottom) shows that the resulting Gaussian distribution gm-SGM represents the data better. This consideration shows shows that SGM is not robust to calibration uncertainty and requires calibrated camera parameters of high precision with low uncertainty.

6.2.2 Correlation

For the analysis of correlations of the RPV-methods to specific calibration parameters, a MCS with 50000 iterations is performed with application of variation and uncertainty of all considered calibration parameters at a distance of 10m. A correlation matrix is set up that is based on all variation and calibration parameters and the estimated distances of the different methods. The important parts of the correlation matrix required for this consideration are extracted and presented in Figure 44 and 45

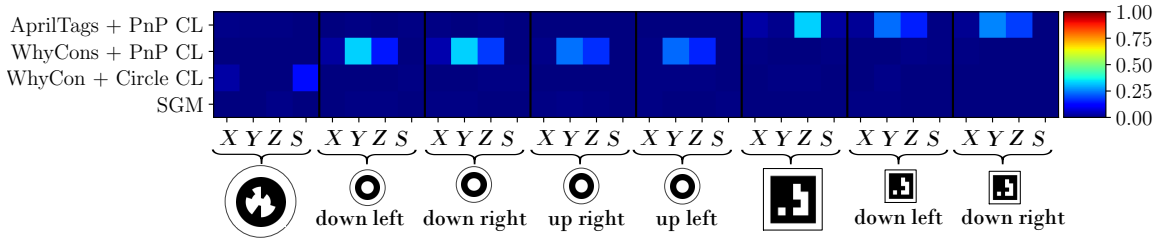


Figure 44: Correlation to uncertainty of the marker pose (App. B.4.4)

by their absolute values. Red indicates a high correlation and dark blue rather non. Please note that only the distance estimation of the left camera for the marker-based methods are considered. The corresponding right camera estimations are shown in Appendix B.4.4.
Figure 44 shows the dependencies of the applied RPV-methods and the calibration uncertainty of the position in X, Y, Z direction and of the marker scaling S. The rotation of each marker is not displayed, because its correlations have shown to be negligible. Figure 44 points out that the PnP-based methods strongly correlate with the calibrated positions in Y- and Z-direction of the markers, which scale the most important reference lengths of the model. Based on these correlations, the most significant direction of each marker is drawn into Figure 46 (a). In contrast to the PnP methods, WhyCon+Circle shows only a great correlation with the scale of the marker, which is most significant for the estimation with a single marker even in comparison to the X-direction itself.

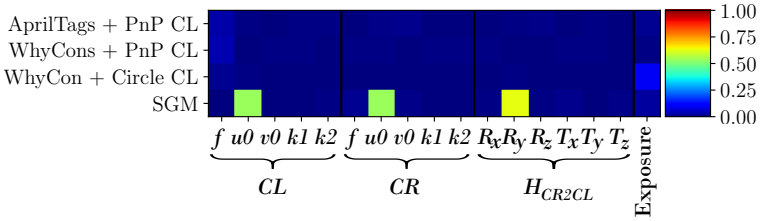
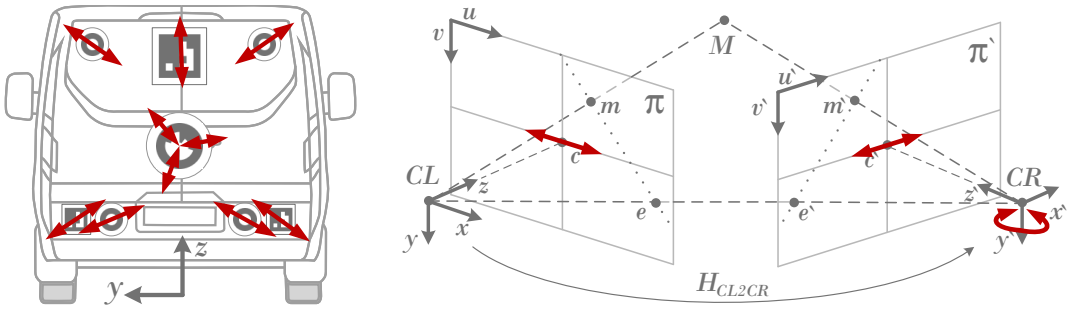


Figure 45: Correlation to camera calibration uncertainty (App. B.4.4)

Figure 45 shows the correlation of the methods with all considered camera calibration parameters. This includes the focal length, principal point and two distortion parameters of each camera as well as the stereo transformation H_{CR2CL}. When considering all marker-based methods, the correlation matrix indicates that only the uncertainty of the focal length has a noticeable influence on the measured distance. In contrast, SGM is highly affected by the uncertainties of the principal points in horizontal direction and also by the uncertainty of the rotation around the y-axis of H_{CR2CL}. Both influences are caused by the direct conjunction with the triangulation calculation. Figure 46 (b) marks out these parameters.
To complete this consideration, Figure 45 also includes a visualization of the methods correlation with image exposure. As pointed out in Section 6.1.3, only WhyCon+Circle has a noticeable correlation. The correlation with the position and orientation trembling of the cameras is negligible and not illustrated.



(a) Marker-based (b) SGM

Figure 46: Visualization of the most influencing calibration parameters (App. B.4.4)

6.2.3 Marker Uncertainty

In Figure 47, the investigations of different severities of uncertainty of the marker calibration are illustrated. Therefore, the standard deviations σ_m of all components of the marker calibration are scaled by a factor s_m. The camera calibration uncertainties stay unchanged. A difference is recognizable for the uncertainties of the measured distances for the marker-based methods between $s_m = 0$ that states no marker calibration and the normal case $s_m = 1$. This tremendous difference confirms the results of the observations of the previous section that the calibration of the markers is most significant for these methods. When increasing the factor up to three, which implies a marker uncertainty of 3cm translation on each axis and 3° rotation around each axis, the uncertainty of the PnP-based methods exceed those of the SGM-approach. This shows the importance of a good calibration of the attachment of the markers. However, in a train application it can be assumed that the markers are calibrated with high precision, which corresponds to a scale better than $s_m = 1$.

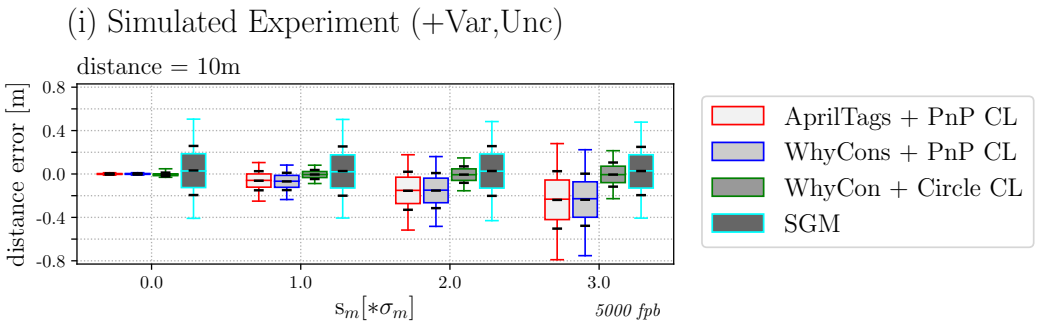


Figure 47: Uncertainty of marker calibration parameters (App. B.4.5)

6.2.4 Camera Uncertainty

Figure 48 examines different severities of uncertainty of the camera calibration. The standard deviations of all components of the camera calibration are scaled by a factor s_c. The marker calibration uncertainties stay unchanged. As discovered in Section 6.2.1 the camera uncertainty has a negligible effect in comparison to the marker uncertainty for

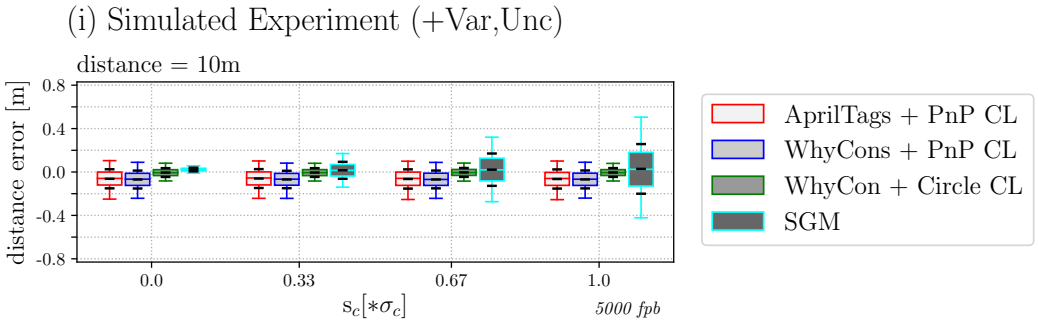


Figure 48: Uncertainty of camera calibration parameters (App. B.4.6)

the marker-based approaches. In contrast, the resulting uncertainty of SGM decreases almost linearly with a smaller camera uncertainty. In order to be competitive with the marker-based methods, SGM needs a camera calibration that is three times less uncertain than the applied parameters.

6.2.5 Influence of the Baseline

An alternative to a better camera calibration is to increase the length of the baseline. When considering Formula 6 (p.11), it is recognizable that a greater baseline increases the disparity for the same distance. This leads to smaller steps in distance for one pixel step in disparity space. Figure 49 applies a Monte-Carlo-Simulation for four different

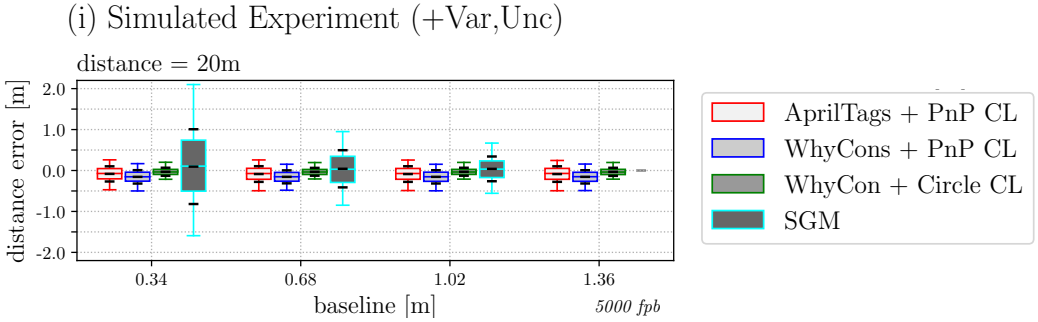


Figure 49: Influence of the baseline on uncertainties (App. B.4.7)

baselines of the used camera system to investigate its influence on the uncertainty of the SGM distance estimation. The standard baseline of the camera system is 0.34m, which is represented by the first box plot of the plot. Then, when linearly increasing the baseline, the uncertainty of SGM decreases with a root shape. At a baseline three times larger than the applied one, the uncertainty of SGM is comparable to the other marker-based methods. Thus, this experiments shows the advantage of great baseline. However, at a baseline of 1.36m the disparity of the object at a distance of 20m exceeds the maximum set disparity of 128px. The maximum disparity is important to limit required resources and computational power. Thus, the matching fails and a distance estimate that is not within the plotted limits of the distance error.
Besides the baseline, the focal length could also be increased, which is not considered since the given focal length of around 12.5mm [DLRStereo] is already rather big.

6.3 Accumulation of RPV-Methods

This section briefly investigates the potential of combining different RPV-methods. Therefore, the correlation of the methods is examined and a weighted aggregation of the different distances based on their estimated uncertainty from simulation is tested.

6.3.1 Correlation of RPV-Methods

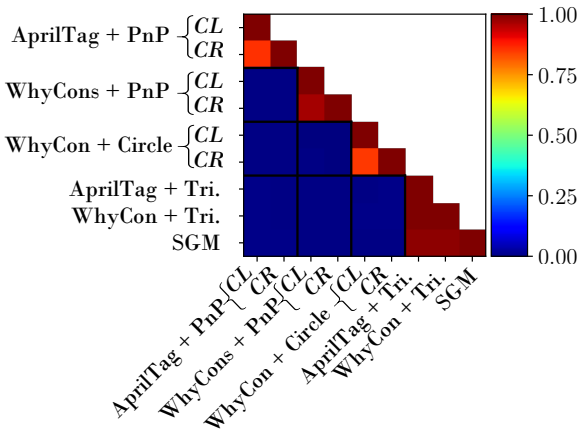


Figure 50: Correlation of Methods

Figure 50 visualizes the correlation of the different RPV-methods. For this correlation analysis, the same extensive Monte-Carlo-Experiment is used as in Section 6.2.2. Figure 50 shows a clear independence between the marker-based methods and the triangulation methods. The stereo methods strongly correlate with each other, caused by the shared correlation to the camera calibration parameters. The monocular marker-based methods show a strong correlation between the estimations of the left *CL* and the right camera *CR*, which is caused by the shared strong correlation to the calibration uncertainty of the markers.

6.3.2 Combination of RPV-Methods

Figure 51 shows an attempt to combine the results of all considered RPV-methods that are applicable in a stereo-camera system with one large attached WhyCon on the vehicle. This includes WhyCon+Circle for *CL* and *CR*, triangulation of the pattern and the application of the SGM approach. There are different simple methods to combine estimations based on a given uncertainty. For instance, only the method with the best resulting uncertainty could be used or the results could be averaged, weighted by their estimated variance. For this theoretical investigation, the uncertainties are extracted from the MCS of Figure 51 (i) since the uncertainties are not calculated on the fly by an error propagation.
Figure 51 (i,left) shows that the estimated uncertainties of the monocular estimations are nearly equal, which also applies for the stereo-based methods. This is caused by the strong correlation to the marker- or the camera uncertainties. At the same time, the

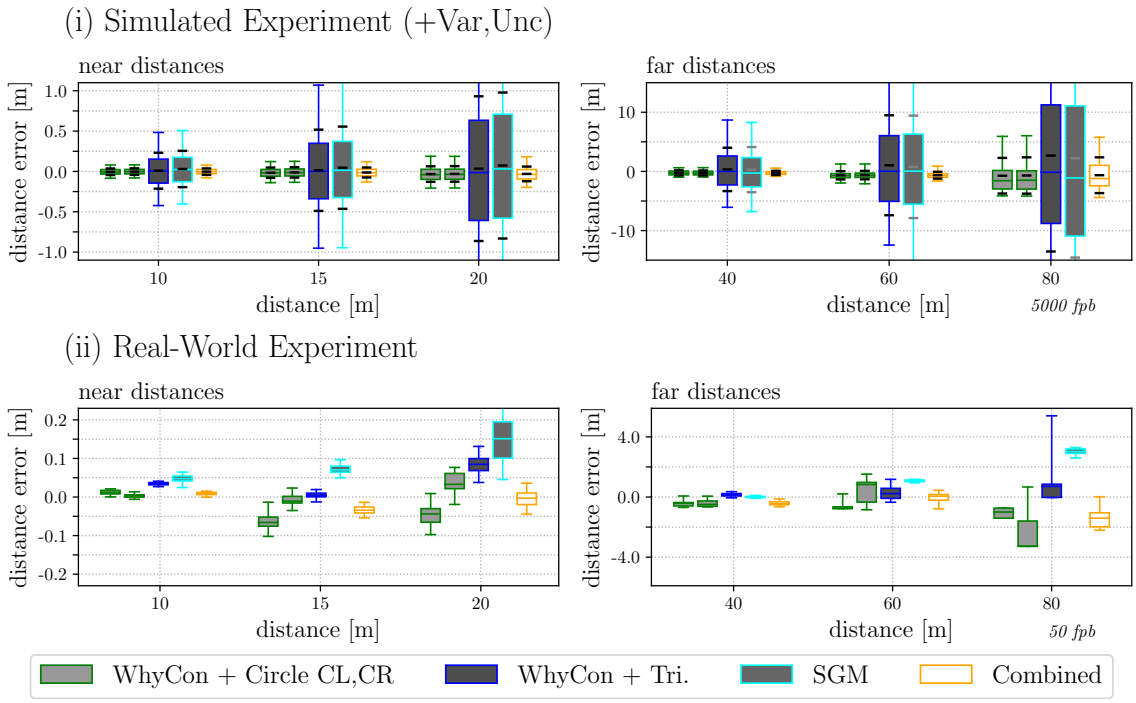


Figure 51: Combination of results based on uncertainties (App. B.4.8)

uncertainties of the stereo-based methods are much higher than those of the monocular-based methods. Thus, it is unattractive for this theoretical investigation to use only the result of the method with the smallest uncertainty. The distance of the *Combined* method is estimated for each iteration based on the uncertainties of Figure 51 (i), formulated in the following equation:

$$d_c = \frac{1}{K} \sum_{m=1}^{4} \frac{1}{\sigma_m^2} d_m \ , \qquad \text{with} \quad K = \sum_{m=1}^{4} \sigma_m^2 \tag{13}$$

Figure 51 (ii) shows a small improvement in accuracy in comparison to the stand alone methods, especially when considering a distance of 20m. However, when considering the resulting uncertainty of the combined method in Figure 51 (i), it is striking that it does not get smaller than the uncertainties of the marker-based methods. This is caused by the shared strong correlation to the calibration uncertainties of the standalone methods. As a result, the uncertainty does not improve.
This theoretical consideration shows that the uncertainty of the marker calibration does not only worsen the estimation for one camera, it also prevents a multi-camera system from reducing the uncertainty. Furthermore, because of the strong influence of the camera calibration on the stereo-based methods, the variance-based weight of the stereo-based method is too small to improve the result based on Equation 13.
However, a multi-camera system comes along with redundant measurements, which ensure a valid result in the case that the WhyCon marker is not detected. Section 6.1.2 shows that this frequently applies for large distances and acute view-angles, whereas SGM succeeds in all considered cases. Thus, the second pattern detection with the right camera and the additional estimation based on SGM increase the robustness.

7 Conclusion

In this work, the application of fiducial marker for relative distance estimation was investigated with particular reference to "Virtual Coupling" of trains. Different marker configurations were evaluated, which include the application of multiple AprilTags as well as four WhyCon to apply a PnP method and a single WhyCon to estimate the distance based on its outer circle. They were compared with an SGM approach and were tested in a stereo setup. The related experiments were conducted in simulation and real-world.
The first research question concerns the overall performance of different marker configurations with respect to the applied distance, view-angle and image exposure and noise. The application of multiple markers provided accurate results and bypassed the dependency to image exposure that occurred when using single markers. However, the occupied marker area was relatively large with a comparatively low application range. In contrast, the single large WhyCon marker was more affected by image exposure, but provided the longest application range, boosted by the fact that its code did not need to be extracted in order to detect the marker. The SGM approach has shown to be superior with respect to the application range even for acute view-angles, while providing the same accuracy and invariance to image exposure.
The second research question addresses the uncertainty of the individual estimations caused by uncertainty in calibration and the correlation to specific calibration parameters. The research has shown that SGM comes with many strong dependencies to camera calibration parameters, which resulted in comparatively large uncertainties. The experiments showed that the uncertainty of the given camera calibration parameters needs to be improved by a factor of three to provide a comparable uncertainty to the other methods. Enlarging the baseline by the same factor provided similar results. The approaches that are based on multiple markers correlated strongly with the calibration of the markers, but less with the camera calibration. The estimation based on the single marker was the most independent to all calibration parameters, which lead to the smallest uncertainty of all considered methods.
The last research question scrutinizes the application of the marker-based methods in a stereo setup. Therefore, the single WhyCon configuration was used to estimate the distance from both cameras individually and by triangulation. The SGM approach was applied to generate a fourth estimation. The individual results were combined based on the estimated uncertainties by the Monte-Carlo-Simulation. The research has shown that the result of the mono-camera method was not improved significantly, due to strong correlations between both single-camera estimations and the low influence of the stereo methods caused by their high uncertainty. However, the robustness was increased in cases of non-detection of the marker due to additional measurements.
The presented results imply that the single WhyCon estimation is most suited for the task of relative position estimation of vehicles in comparison to all considered methods, based on the given camera setup and used calibration parameters. Therefore, I suggest using a stereo-camera system with SGM to ensure robustness and apply fiducial markers to gain high certainty. Effort should be put into the geometrical calibration. Since the markers are robustly detected for short distances, the baseline should be set as large as possible.

8 Discussion and Outlook

This study took advantage of existing implementations of methods and considered as well as verified conclusions of other literature. It used proven evaluation methods to construct, apply and evaluate various marker configurations for train applications. The proposed evaluation pipeline allows a versatile comparison of the applied methods with respect to variation of the scene and calibration uncertainty.
Concerning the variation of the scene, the influence of image exposure on fiducial markers has shown to be a crucial factor. This has been experimentally considered in detail in (Mangelson et al., 2016). Comparable to their work or related (Bergamasco et al., 2016; Birdal et al., 2016), this was circumvented by using multiple circular tags. However, the application of a single WhyCon marker that is attached with a binary code similar to WhyCode has shown to be most appropriate. That‘s because it possesses great independence to calibration uncertainty and a long application range. Its accuracy could be increased by applying a subsequent sub-pixel edge detection and ellipse fitting (Cucci, 2016). In the case the code is omitted, the ratio of the inner and outer circle can be used to enhance the estimation (Krajník et al., 2014).
The consideration of uncertainty revealed weaknesses of stereo-based approaches such as SGM, because of their strong sensitivity to uncertainty in camera calibration. The resulting high uncertainty of stereo methods showed to be disadvantageous when trying to combine multiple methods. However, this observation is strongly depending on the uncertainties of calibration parameters that are applied in this study. Different calibration methods and different camera setups can increase the calibrated uncertainties and will greatly change the applicability of stereo methods. This was shown exemplary by increasing the baseline of the camera system.
The evaluation of this study was mainly based on simulation and substantiated acquired conclusions by real-world experiments. This implies that all conclusions are limited by the possibilities and characteristics of the simulator. The image quality was considered closely in terms of anti-aliasing. In this study, only ambient light was applied, which allows a precise analysis of exposure effects. Critical to consider is the missing consideration of image blooming since (Mangelson et al., 2016) has shown its non-negligible influence on marker-based estimations. Motion blur is not supported either and was not considered in this work. It has shown in many applications to have a substantial influence on markers, e.g. in (Calvet et al., 2016). In applications of relative position estimation of vehicles, such as "Virtual Coupling" of trains, this effect is reduced, because the image position of the preceding vehicle is usually not changing rapidly. But the wiggling of the train could cause substantial image blurring, which should be investigated in a separate study. Also, the pollution of the markers was not considered and only one vehicle type was used during the general evaluation.
The proposed evaluation pipeline allows to easily evaluate and compare versatile methods that compute the distance to a specific object. This framework and the associated simulator will be developed further to increase the number of possible applications. In the context of relative position estimation of vehicles, the pipeline will be expanded to evaluate tracking-based methods, which could also increase the accuracy and reliability of the marker-based methods. Furthermore, different light effects such as shadowing and reflection will be considered to make the variation of the scene even more versatile.

References

Ababsa, Fakhr-eddine and Malik Mallem (2004). "Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems". In: *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*. Ed. by Judith Brown. New York, NY: ACM, p. 431. ISBN: 1581138849. DOI: 10.1145/1044588.1044682.

Akenine-Möller, Tomas, Eric Haines, and Naty Hoffman (2008). *Real-Time Rendering 3rd Edition*. Natick, MA, USA: A. K. Peters, Ltd. ISBN: 987-1-56881-424-7.

Badino, Hernán, Uwe Franke, and Rudolf Mester (2007). "Free Space Computation Using Stochastic Occupancy Grids and Dynamic Programming". In: *Proc. Int'l Conf. Computer Vision, Workshop Dynamical Vision*.

Badino, Hernán, Uwe Franke, and David Pfeiffer (2009). "The Stixel World - A Compact Medium Level Representation of the 3D-World". In: URL: http://www.lelaps.de/papers/badino_dagm09.pdf.

Bergamasco, Filippo et al. (2011). "RUNE-Tag: A high accuracy fiducial marker with strong occlusion resilience". In: *CVPR 2011*. IEEE, pp. 113–120. ISBN: 978-1-4577-0394-2. DOI: 10.1109/CVPR.2011.5995544.

Bergamasco, Filippo, Andrea Albarelli, and Andrea Torsello (2013). "Pi-Tag: A fast image-space marker design based on projective invariants". In: *Machine Vision and Applications* 24.6, pp. 1295–1310. ISSN: 0932-8092. DOI: 10.1007/s00138-012-0469-6.

Bergamasco, Filippo et al. (2016). "An Accurate and Robust Artificial Marker Based on Cyclic Codes". In: *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* 38.12, pp. 2359–2373. DOI: 10.1109/TPAMI.2016.2519024.

Bernini, Nicola et al. (2014). "Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey". In: *IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), 2014*. Piscataway, NJ: IEEE, pp. 873–878. ISBN: 978-1-4799-6078-1. DOI: 10.1109/ITSC.2014.6957799.

Birdal, Tolga, Ievgeniia Dobryden, and Slobodan Ilic (2016). "X-Tag: A Fiducial Tag for Flexible and Accurate Bundle Adjustment". In: *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, pp. 556–564. ISBN: 978-1-5090-5407-7. DOI: 10.1109/3DV.2016.65.

Boyat, Ajay Kumar and Brijendra Kumar Joshi (2015). "A Review Paper: Noise Models in Digital Image Processing". In: *Signal & Image Processing : An International Journal* 6.2, pp. 63–75. ISSN: 22293922. DOI: 10.5121/sipij.2015.6206.

Britto, Joao et al. (2015). "Model identification of an unmanned underwater vehicle via an adaptive technique and artificial fiducial markers". In: *OCEANS 2015 - MTS/IEEE Washington*. Piscataway, NJ: IEEE, pp. 1–6. ISBN: 978-0-9339-5743-5. DOI: 10.23919/OCEANS.2015.7404391.

Brown, Duane C. (1971). "Close-range camera calibration". In: *PHOTOGRAMMETRIC ENGINEERING* 37.8, pp. 855–866.

Calvet, Lilian et al. (2016). "Detection and Accurate Localization of Circular Fiducials under Highly Challenging Conditions". In: *29th IEEE Conference on Computer*

Vision and Pattern Recognition. Piscataway, NJ: IEEE, pp. 562–570. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.67.

Caraffi, Claudio et al. (2012). "A system for real-time detection and tracking of vehicles from a single car-mounted camera". In: *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, pp. 975–982. ISBN: 978-1-4673-3063-3. DOI: 10.1109/ITSC.2012.6338748.

Chen, Shi-Huang and Ruie-Shen Chen (2011). "Vision-Based Distance Estimation for Multiple Vehicles Using Single Optical Camera". In: *Second International Conference on Innovations in Bio-inspired Computing and Applications (IBICA), 2011*. Piscataway, NJ: IEEE, pp. 9–12. ISBN: 978-1-4577-1219-7. DOI: 10.1109/IBICA.2011.7.

Cordts, Marius et al. (2017). "The Stixel world: A medium-level representation of traffic scenes". In: *Image and Vision Computing*. ISSN: 02628856. DOI: 10.1016/j.imavis.2017.01.009.

Cucci, D. A. (2016). "Accurate Optical Target Pose Determination for Application in Aerial photogrammetry". In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* III-3, pp. 257–262. ISSN: 2194-9050. DOI: 10.5194/isprsannals-III-3-257-2016.

Danescu, Radu and Sergiu Nedevschi (2014). "A Particle-Based Solution for Modeling and Tracking Dynamic Digital Elevation Maps". In: *IEEE Transactions on Intelligent Transportation Systems* 15.3, pp. 1002–1015. ISSN: 1524-9050. DOI: 10.1109/TITS.2013.2291447.

Dhanaekaran, Surender et al. (2015). "A Survey on Vehicle Detection based on Vision". In: *Modern Applied Science* 9.12, p. 118. ISSN: 1913-1852. DOI: 10.5539/mas.v9n12p118.

DLR, ed. (2016). *Im Hochgeschwindigkeitszug durch die Nacht - DLR Wissenschaftler entwickeln Zug-zu-Zug-Kommunikation*. Germany. URL: http://www.dlr.de/dlr/desktopdefault.aspx/tabid-10122/333_read-17514/#/gallery/22712.

Elfes, Alberto (1989). "Using occupancy grids for mobile robot perception and navigation - Computer". In: *IEEE*.

Erbs, Friedrich, Alexander Barth, and Uwe Franke (2011). "Moving vehicle detection by optimal segmentation of the Dynamic Stixel World". In: *IEEE Intelligent Vehicles Symposium (IV), 2011 ; 5 - 9 June 2011 ; Baden-Baden, Germany*. Piscataway, NJ: IEEE, pp. 951–956. ISBN: 978-1-4577-0890-9. DOI: 10.1109/IVS.2011.5940532.

Ernst, Ines and Heiko Hirschmüller (2008). "Mutual Information Based Semi-Global Stereo Matching on the GPU". In: *Advances in visual computing*. Ed. by George Bebis. Vol. 5358. Lecture Notes in Computer Science. Berlin: Springer, pp. 228–239. ISBN: 978-3-540-89638-8. DOI: 10.1007/978-3-540-89639-5{\textunderscore}22.

Fiala, Mark (2005). "ARTag, a Fiducial Marker System Using Digital Techniques". In: *CVPR '05 Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 590–596.

Fischler, Martin A. and Robert C. Bolles (1981). "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6, pp. 381–395. ISSN: 00010782. DOI: 10.1145/358669.358692.

Forster, Roger (2000). "Manchester encoding: opposing definitions resolved". In: *Engineering Science and Education Journal*.

Funk, Eugen (2017). *Next Generation Train. Meilenstein 24301703*. Ed. by Deutsches Zentrum für Luft- und Raumfahrt e.V. Berlin.

Gatrell, Lance B. and William A. Hoff (1991). "Robust Image Features: Concentric Contrasting Circles and Their Image Extraction". In: *Proceedings Volume 1612, Cooperative Intelligent Robotics in Space II* 1992.

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Griesbach, Denis, Dirk Baumbach, and Sergey Zuev (2014). "Stereo-vision-aided inertial navigation for unknown indoor and outdoor environments". In: *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Piscataway, NJ: IEEE, pp. 709–716. ISBN: 978-1-4673-8054-6. DOI: 10.1109/IPIN.2014.7275548.

Heikkila, J. and O. Silven (1997). "A four-step camera calibration procedure with implicit image correction". In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Comput. Soc, pp. 1106–1112. ISBN: 0-8186-7822-4. DOI: 10.1109/CVPR.1997.609468.

Hermann, Simon and Reinhard Klette (2013). "Iterative Semi-Global Matching for Robust Driver Assistance Systems". In: *Computer Vision - ACCV 2012*. Ed. by David Hutchison et al. Vol. 7726. Lecture Notes in Computer Science / Image Processing, Computer Vision, Pattern Recognition, and Graphics. Berlin/Heidelberg: Springer Berlin Heidelberg, pp. 465–478. ISBN: 978-3-642-37430-2. DOI: 10.1007/978-3-642-37431-9{\textunderscore}36.

Hirschmuller, H. (2005). "Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information". In: *CVPR*. Ed. by Cordelia Schmid. Los Alamitos, Calif.: IEEE Computer Society, pp. 807–814. ISBN: 0-7695-2372-2. DOI: 10.1109/CVPR.2005.56.

Hirschmüller, Heiko (2007). "Stereo Processing by Semi-Global Matching and Mutual Information". In: *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*. URL: https://core.ac.uk/download/pdf/11134866.pdf.

Jähne, Bernd (2005). *Digitale Bildverarbeitung*. 6th ed. Springer Berlin Heidelberg. ISBN: 3-540-24999-0.

JCGM (2008a). "Evaluation of measurement data - Guide to the expression of uncertainty in measurement: GUM 1995 with minor corrections". In: *Joint Committee for Guides in Metrology*.

– (2008b). "Evaluation of measurement data - Supplement 1 to the Guide to the expression of uncertainty in measurement: Propagation of distributions using a Monte Carlo method". In: *Joint Committee for Guides in Metrology*.

Kato, H. and M. Billinghurst (1999). "Marker tracking and HMD calibration for a video-based augmented reality conferencing system". In: *Proceedings, 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. Los Alamitos, Calif: IEEE Computer Society, pp. 85–94. ISBN: 0-7695-0359-4. DOI: 10.1109/IWAR.1999.803809.

Krajník, Tomáš et al. (2013). "External localization system for mobile robotics". In: *16th International Conference on Advanced Robotics (ICAR), 2013*. Piscataway, NJ: IEEE, pp. 1–6. ISBN: 978-1-4799-2722-7. DOI: 10.1109/ICAR.2013.6766520.

Krajník, Tomáš et al. (2014). "A Practical Multirobot Localization System". In: *Journal of Intelligent & Robotic Systems* 76.3-4, pp. 539–562. ISSN: 0921-0296. DOI: 10.1007/s10846-014-0041-x.

Lehmann, Florian (2015). *Evaluierung eines Inertialsensors. Implementierung einer virtuellen Kamera mit Verzeichnung.* Ed. by Deutsches Zentrum für Luft- und Raumfahrt e.V. in der Helmholtz-Gemeinschaft.

– (2016). *Implementierung einer virtuellen Stereokamera.* Ed. by Deutsches Zentrum für Luft- und Raumfahrt e.V. in der Helmholtz-Gemeinschaft.

Lenz, Philip et al. (2011). "Sparse scene flow segmentation for moving object detection in urban environments". In: *IEEE Intelligent Vehicles Symposium (IV), 2011 ; 5 - 9 June 2011 ; Baden-Baden, Germany*. Piscataway, NJ: IEEE, pp. 926–932. ISBN: 978-1-4577-0890-9. DOI: 10.1109/IVS.2011.5940558.

Lepetit, Vincent, Francesc Moreno-Noguer, and Pascal Fua (2009). "EPnP: An Accurate O(n) Solution to the PnP Problem". In: *International Journal of Computer Vision* 81.2, pp. 155–166. ISSN: 0920-5691. DOI: 10.1007/s11263-008-0152-6.

Lessmann, Stephanie et al. (2016). "Probabilistic distance estimation for vehicle tracking application in monocular vision". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 1199–1204. ISBN: 978-1-5090-1821-5. DOI: 10.1109/IVS.2016.7535542.

Ley, Andreas, Ronny Hänsch, and Olaf Hellwich (2016). "SyB3R: A Realistic Synthetic Benchmark for 3D Reconstruction from Images". In: *SpringerLink*.

Li, Shiqi, Chi Xu, and Ming Xie (2012). "A Robust O(n) Solution to the Perspective-n-Point Problem". In: *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* 34.7, pp. 1444–1450. DOI: 10.1109/TPAMI.2012.41.

Lightbody, Peter, Tomas Krajnik, and Marc Hanheide (2017). "A Versatile High-Performance Visual Fiducial Marker Detection System with Scalable Identity Encoding". In: *Proceedings of the Symposium on Applied Computing*, pp. 276–282.

Liu, Yinan et al. (2017). "Calculating Vehicle-to-Vehicle Distance Based on License Plate Detection". In: *Advances in Intelligent Systems and Computing* 454.

Lu, Yin-Yu et al. (2011). "A vision-based system for the prevention of car collisions at night". In: *Machine Vision and Applications* 22.1, pp. 117–127. ISSN: 0932-8092. DOI: 10.1007/s00138-009-0239-2.

Lucas, Bruce D. and Takeo Kanade (1981). "An iterative image registration technique with an application to stereo vision". In: *In IJCAI81*, pp. 674–679.

Mangelson, Joshua G. et al. (2016). "Robust visual fiducials for skin-to-skin relative ship pose estimation". In: *OCEANS 2016 MTS/IEEE Monterey*. IEEE, pp. 1–8. ISBN: 978-1-5090-1537-5. DOI: 10.1109/OCEANS.2016.7761168.

Menze, Moritz and Andreas Geiger (2015). "Object scene flow for autonomous vehicles". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, pp. 3061–3070. ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298925.

Moratto, Zack (2013). *Semi-Global Matching*. Ed. by LUNOKHOD. URL: http://lunokhod.org/?p=1356.

Naimark, L. and E. Foxlin (2002). "Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker". In: *Proceedings / International Symposium on Mixed and Augmented Rality*. Los Alamitos, Calif.: IEEEE Computer Society, pp. 27–36. ISBN: 0-7695-1781-1. DOI: 10.1109/ISMAR.2002.1115065.

Nakamura, Katsuyuki et al. (2013). "Real-time monocular ranging by Bayesian triangulation". In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 1368–1373. ISBN: 978-1-4673-2755-8. DOI: 10.1109/IVS.2013.6629657.

Olson, Edwin (2011). "AprilTag: A robust and flexible visual fiducial system". In: *2011 IEEE International Conference on Robotics and Automation*. Ed. by Antonio Bicchi. Piscataway, NJ: IEEE, pp. 3400–3407. ISBN: 978-1-61284-386-5. DOI: 10.1109/ICRA.2011.5979561.

Oniga, F. and S. Nedevschi (2010). "Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle, and Obstacle Detection". In: *IEEE Transactions on Vehicular Technology* 59.3, pp. 1172–1182. ISSN: 0018-9545. DOI: 10.1109/TVT.2009.2039718.

Park, Ki-Yeong and Sun-Young Hwang (2014). "Robust range estimation with a monocular camera for vision-based forward collision warning system". In: *TheScientificWorldJournal* 2014, p. 923632. ISSN: 1537-744X. DOI: 10.1155/2014/923632.

Pertile, Marco et al. (2015). "Uncertainty evaluation of a vision system for pose measurement of a spacecraft with fiducial markers". In: *Metrology for Aerospace, IEEE* 2015.

Ponte Muller, Fabian de (2017). "Survey on Ranging Sensors and Cooperative Techniques for Relative Positioning of Vehicles". In: *Sensors (Basel, Switzerland)* 17.2. ISSN: 1424-8220. DOI: 10.3390/s17020271.

Quan, Long and Zhongdan Lan (1999). "Linear N-point camera pose determination". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.8, pp. 774–780. ISSN: 01628828. DOI: 10.1109/34.784291.

Remondino, Fabio et al. (2013). "Dense image matching: Comparisons and analyses". In: *2013 Digital Heritage International Congress (DigitalHeritage)*. IEEE, pp. 47–54. ISBN: 978-1-4799-3170-5. DOI: 10.1109/DigitalHeritage.2013.6743712.

Schreer, Oliver (2005). *Stereoanalyse und Bildsynthese: Mit 6 Tabellen*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg. ISBN: 3-540-23439-X. DOI: 10.1007/3-540-27473-1. URL: http://dx.doi.org/10.1007/3-540-27473-1.

Seng, Kian Lee et al. (2013). "Vision-based State Estimation of an Unmanned Aerial Vehicle". In: *Trends in Bioinformatics* 10, pp. 11–19.

Sivaraman, Sayanan and Mohan M. Trivedi (2013a). "A review of recent developments in vision-based vehicle detection". In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 310–315. ISBN: 978-1-4673-2755-8. DOI: 10.1109/IVS.2013.6629487.

Sivaraman, Sayanan and Mohan Manubhai Trivedi (2013b). "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis". In: *IEEE Transactions on Intelligent Transportation Systems* 14.4, pp. 1773–1795. ISSN: 1524-9050. DOI: 10.1109/TITS.2013.2266661.

Stein, G. P., O. Mano, and A. Shashua (2003). "Vision-based ACC with a single camera: bounds on range and range rate accuracy". In: *Proceedings / IEEE IV 2003,*

Intelligent Vehicles Symposium. Piscataway, NJ: IEEE Operations Center, pp. 120–125. ISBN: 0-7803-7848-2. DOI: 10.1109/IVS.2003.1212895.

Stein, Gideon P., D. Ferenez, and Ofer Avni (2012). "Estimating distance to an object using a sequence of images recorded by a monocular camera". Pat. US8164628 B2.

Thoman, Peter (2014). *Diving into Anti-Aliasing: Sampling-based Anti-Aliasing Techniques*. Ed. by Beyond3D. URL: https://www.beyond3d.com/content/articles/122/4.

Thrun, Sebastian, Wolfram Burgard, and Dieter Fox (2006). *Probabilistic robotics*. Intelligent robotics and autonomous agents series. Cambridge, Mass.: MIT Press. ISBN: 978-0-262-20162-9.

Tukey, John W. (1977). "Exploratory data analysis". In: *Addison-Wesley*, pp. 530–537.

Urban, Steffen, Jens Leitloff, and Stefan Hinz (2016). "MLPnP - A Real-Time Maximum Likelihood Solution to the Perspective-n-Point Problem". In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* III-3, pp. 131–138. ISSN: 2194-9050. DOI: 10.5194/isprs-annals-III-3-131-2016.

Walters, Austin and Bhargava Manja (2015). "ChromaTag - A Colored Fiducial Marker". In: *International Conference on Computer Vision* arXiv:1708.02982.

Wang, John and Edwin Olson (2016). "AprilTag 2: Efficient and robust fiducial detection". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4193–4198. ISBN: 978-1-5090-3762-9. DOI: 10.1109/IROS.2016.7759617.

Wilson, Daniel B., Ali H. Goktogan, and Salah Sukkarieh (2014). "A vision based relative navigation framework for formation flight". In: *IEEE International Conference on Robotics and Automation (ICRA), 2014*. Piscataway, NJ: IEEE, pp. 4988–4995. ISBN: 978-1-4799-3685-4. DOI: 10.1109/ICRA.2014.6907590.

Winkens, Christian and Dietrich Paulus (2017). "Long Range Optical Truck Tracking". In: *Proceedings of the 9th International Conference on Agents and Artificial Intelligence*. SCITEPRESS - Science and Technology Publications, pp. 330–339. ISBN: 978-989-758-219-6. DOI: 10.5220/0006296003300339.

Zhang, Hongmou et al. (2017). "Uncertainty Model for Template Feature Matching". In: *PSIVT2017*.

Zhang, Z. (2000). "A flexible new technique for camera calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11, pp. 1330–1334. ISSN: 01628828. DOI: 10.1109/34.888718.

Technology List

[OSG] OpenSceneGraph-3.4.0. 2015. *OpenSceneGraph is an OpenGL-based high performance 3D graphics toolkit for visual simulation, games, virtual reality, scientific visualization, and modelin*. `http://www.openscenegraph.org`. Last downloaded 2017-06-12.

[OpenGL] OpenGL. *The Industry's Foundation of High Performance Graphics*. `https://www.opengl.org/`. Last downloaded 2017-06-12. Embedded in OpenSceneGraph.

[OpenCV] OpenCV-3.1 2015. *Open Source Computer Vision Library*. `http://opencv.org/`. Last downloaded 2016.

[OSLib] OSLib. DLR Intern. *C++ Software Library for Image Processing*. Implements basics structures, classes and algorithms. Last downloaded 2017-07-27.

[OSVisionLib] OSVisionLib. DLR Intern. *C++ Software Library for Image Processing*. Implements computer vision algorithms and interfaces to access external libraries. Last downloaded 2017-07-27.

[AprilTagLib] Michael Kaess. 2012. *AprilTags Library*. `https://github.com/NifTK/apriltags`. Last downloaded 2016.

[WhyConLib] Tomáš Krajník, Matias Nitsche, Jan Faigl. 2016. *WhyCon*. `https://github.com/LCAS/whycon`. Last downloaded Arpil 2017.

[CalLab] K. H. Strobl and W. Sepp and S. Fuchs and C. Paredes and M. Smisek and K. Arbter. *DLR CalDe and DLR CalLab*. Institute of Robotics and Mechatronics, German Aerospace Center (DLR). Oberpfaffenhofen, Germany. `http://www.robotic.dlr.de/callab/`. Last checked 2017.

[DLRStereo] DLR. *Outdoor Stereo Camera*. Cameras: Prosilica GC1380H (resolution: 1360x1024, cell-size:$6.5^2 \mu m^2$). Baseline: 0.34m.

[DellPrecision] Dell Precision Tower 3620. Processor: Intel(R) Xeon(R) CPU E3-1270 v5 @ 3.6 HHz 4 Cores 8 Threads. Graphic Card: NVIDIA Quadro M4000 8 GB GDDR5 1664 CUDA Cores.

[Blend1] Mesh of a train. Blendswap. *German Train BR646 of the UBB*. Source and license information: `https://www.blendswap.com/blends/view/83719`. License type: CC-BY. Last downloaded 2017-06-25. Changes: Added DLR logo.

[Blend2] Mesh of a rail. Blendswap. *Train*. Source and license information: `https://www.blendswap.com/blends/view/22626`. License: CC-Zero. Last downloaded 2017-04-23. Changes: Used and changed rails and ground.

[Town] Institut für Verkehrssystemtechnik. DLR Intern. *Demo Small Town*.

[GLM-80] Bosch. *Bosch GLM-80*.

Appendix

The appendix provides further considerations and completions that are less decisive for the work itself. This includes an further investigation of the fundamental methods. Also, all parameters used in this experiment are listed and more information to the individual conducted experiments are provided. Finally, a few more simulated experiments are conducted.

A - Method Characteristics

A.1 - PnP Comparison

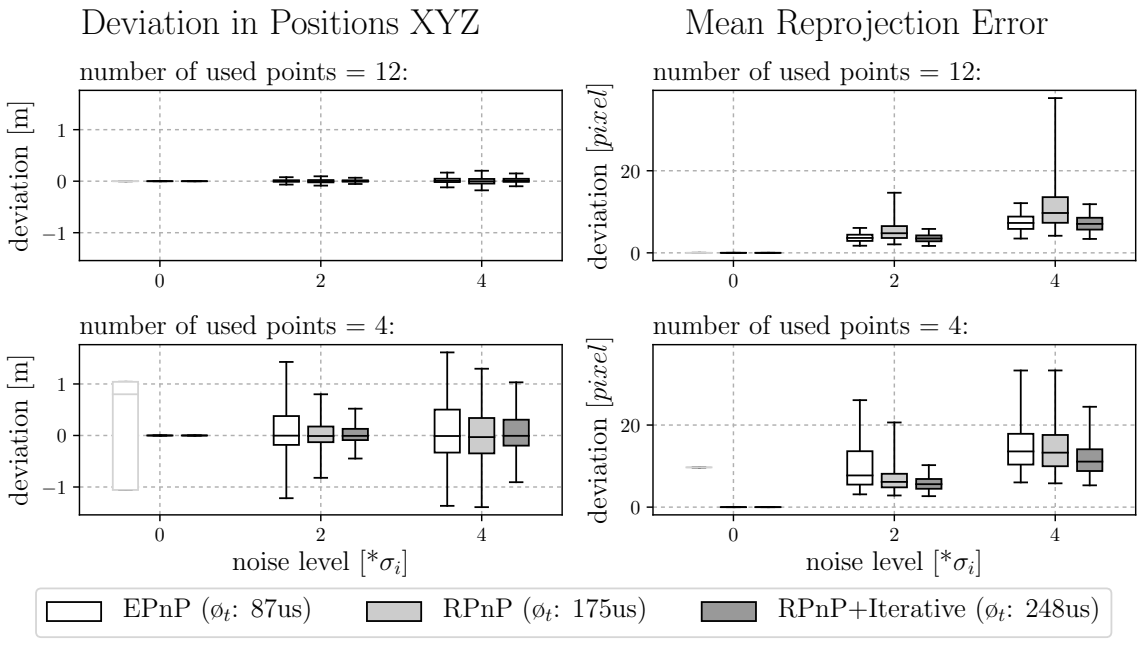
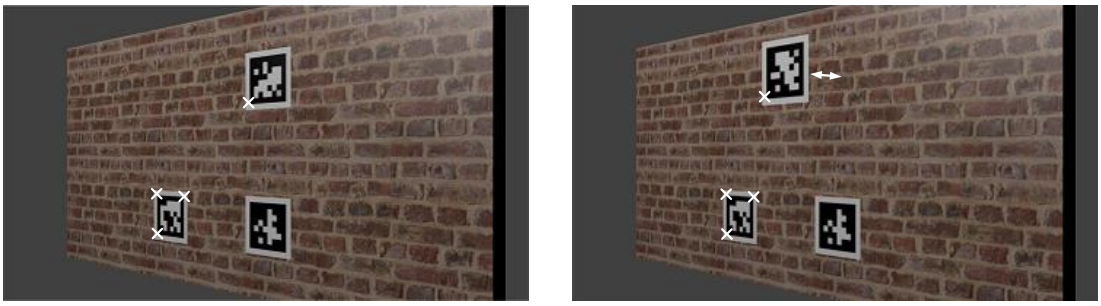


Figure 52: Comparison of PnP-methods based on position deviation and reprojection error of noisy image- and world point correspondences for a scene with non-planar arranged world points.

Figure 53 visualizes the applied camera pose and used correspondences for comparing different PnP-methods. The chosen points are the corners of the visualized AprilTag. However, AprilTags are only used for visualization and its detection is not applied during these experiments. White crosses mark points that are used for the 4-point evaluation. (a) shows the setup for the experiment of Section 3.2.2. All applied world points are co-planar. (b) shows a setup where the upper marker is positioned one half meter in front of the wall, marked with a white arrow. This setup is used in figure 52 to compare EPNP and RPNP for non-planar world points. When using twelve

correspondences EPNP is more accurate than RPNP. However, when using only four points, which is the use case for this work, RPNP is more accurate. Furthermore, in the case of non-noised data EPNP shows again a degenerated solution.



(a) Planar setup (b) Non-planar setup

Figure 53: Visualization of the PnP-test setups

A.2 - Apriltag Characteristics

(a) 1360x1024 pixel (b) 250x350 pixel (c) 1360x1024 pixel

Figure 54: Application of AprilTag pattern detection on an outdoor image (a), a corresponding subimage (b) and an indoor image (c). Blue quads illustrate the detected AprilTags.

This experiment addresses the strong correspondences of the quad detection and identification of AprilTags. Table 3 (c) shows that the method generates multiple candidates of quads, which are subsequently verified by the code identification. Furthermore, the number gets greatly increased in texture rich images, as shown in figure 54 (a). Because of the gravel, a high number of quad candidates is produced, which also increases the false positive rate of detected AprilTags of the used Tag-Set.
Table 3 also maintains the average required time of 100 repetitions of processing each frame. It shows that image (a) requires much more computational time than the indoor image (c). Moreover, the process occupied multiples cores on a [DellPrecision], which shows that the processing of a full image is not real-time capable in the underlying implementation. However, the results of image (c) show that a restriction of the search area in the image, as for instance by tracking the vehicle, would lead to faster process-

ing time. This supports the decision not to evaluate the processing time in this study.

Image	Quads	AprilTags	∅ Time
(a)	348	12	360 ms
(b)	11	3	10 ms
(c)	24	4	135 ms

Table 3: Characteristics of the AprilTag detection for images from figure 54

A.3 - WhyCon Characteristics

A.3.1 - WhyCon Time Characteristics

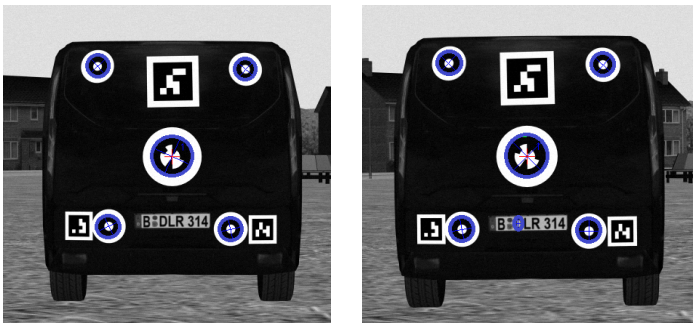
Step	unit	Left Image	Right Image
Image Pyramid + Adaptive Thresholding	*ms*	27.42	27.44
Detection in Image Pyramid	*ms*	164.42	173.71
Redetection	*ms*	0.65	4.66
Detected WhyCons		5	6
Code extraction	*ms*	1.58	1.90
WhyCon Selection and Pose estimation	*ms*	0.0056	0.0063
Remaining WhyCon		1	1

Table 4: Time Characteristics of the applied processing with WhyCon

Table 4 lists the required computational time for each step of the single WhyCon distance estimation based on the stereo frame of Figure 55. The majority of the processing time is consumed by the initial detection of the pattern in the image pyramid, which processes the entire image in multiple levels. The redetection, that represents the processing by the original method, requires only a minor part of the overall computational time. Thus, the processing in the image pyramid could be reduced greatly in the in presence of marker tracking.

A.3.2 - WhyCon Circle Width

Figure 56 shows an experiment to evaluate different widths of the outer circle of the WhyCon pattern. This is done by varying the ratio of the circle width to the inner radius. In the configuration of WhyCons+PnP, a ratio of 0.6 is used that approximately corresponds to the ratio used for a test marker provided in [WhyConLib]. Figure 56 shows that a ratio of 0.5 or less provides a better detection range in the proposed setup based on an adaptive thresholding. This experiment shows that WhyCons+PnP has the potential to provide an even further application range than estimated in Section 6.1.2.



(a) Left image (b) Right image

Figure 55: Illustration of all detected WhyCon pattern in the stereo image

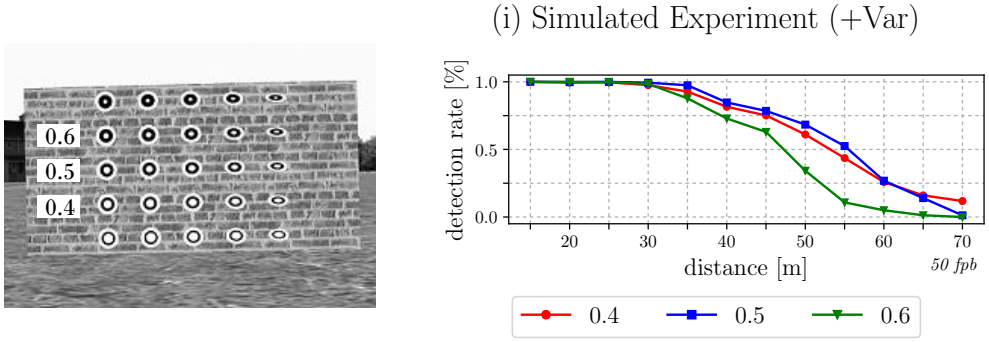


Figure 56: Experiment to find the most suitable width for WhyCon

A.3.3 - WhyCon and Image Noise

For all simulated experiments of this study, image noise is applied based on the noise model of Section 3.1.3 (p.10). Its influence on the methods has shown to be very small, which is why it was not considered separately in this study. However, one characteristic of WhyCon has been noticed, which is outlined in this section. First of all, Figure 57 (i) shows only a weak influence of image noise. It evaluates WhyCon+Circle at a distance of 80m with applied image noise (on) and without (off), while applying the general variation of the simulation. The figure shows the individual estimations for the left and the right camera. The applied image noise does not greatly change the distribution.
In contrast, Figure 57 (ii) shows a different behavior. For this experiment, 200 frames from the same position for the left and the right camera are captured. Thus, they represent two snapshots. The images are captured either with an image exposure of 0.1 and 0.8 for either off and on image exposure. In the case that image noise is applied, it is striking that only for an exposure of 0.1 a spreading occurs. Furthermore, the spreads vary in their severity.
An explanation for this is the hard classification in black and white pixels of WhyCon. In relation, Figure 41 (p.41) (left) illustrates for low image exposure that the value of x_i and the estimated threshold are very close. Thus, if the range of the noise of x_i overlaps with the estimated threshold, the assignment of x_i varies between different images from the same pose. As more border pixels of the projected pattern overlap with the

estimated threshold, as greater becomes the spreading of the resulting distance. This effect also occurs for long exposure as casually illustrated in Figure 41 (p.41) (right), but less likely since the value range is much wider.
This effect is strongly related to the influence of image exposure. (Krajník et al., 2014) propose to use the ratio of the inner and outer circle to correct the estimated results and state that this "compensation of the pattern diameter reduces the average localization error by approximately 15%".

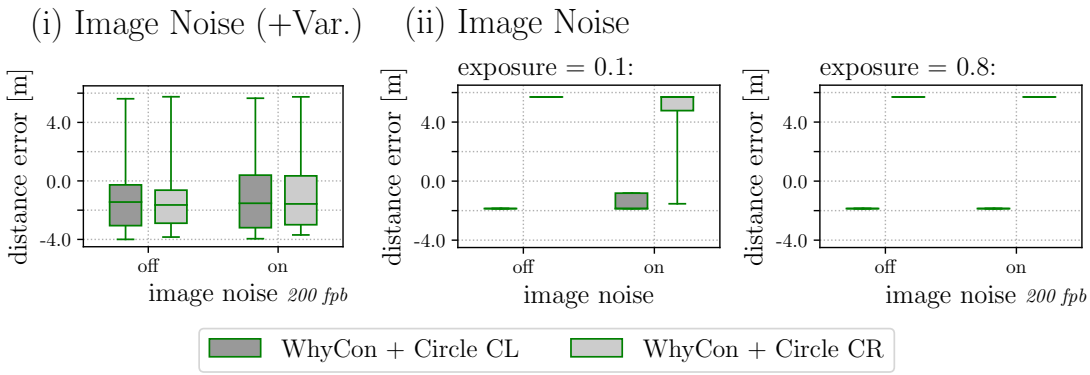
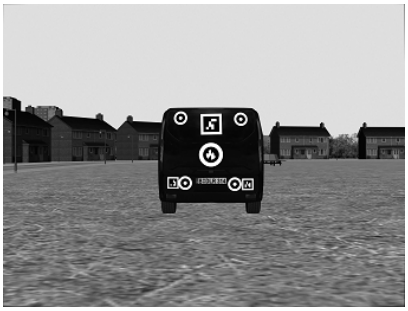


Figure 57: Evaluation of the application of image noise at a distance of 80m

A.4 - Consideration of Time



Step	Time [ms]
AprilTags + PnP CL	182
WhyCons + PnP CL	206
WhyCon + Circle CL	200
SGM	212

Table 5: Exemplary consideration of time

Figure 5 presents a short consideration of the computation time of the four most important methods of this study. The Time is average over 1000 distance estimations with each method based on the pictured image and run on a [DellPrecision]. The results show that all methods require approximately the same time for this exemplary image. Please note that the AprilTag detection uses multiple CPUs, WhyCon runs on one CPU and SGM on the GPU.

B - Tables

B.1 - Variation Parameters

Table 6 shows the specifications of the variation parameters. Each parameter can be overwritten by the individual experiment, especially H_{CL2RP} (see Figure 18, p.21) with the distance d and view-angle α. The trembling of the position is defined by $\sigma_{Tremble}$ that is used to variate $H_{Tremble}$ for each iteration. The sampling on the image exposure is defined to be at least 0.1 to prevent pointless calculations.

(1) Image noise

	noise
$\boldsymbol{N_E}$	0.2658
G	59.1944

(2) Other image effects

	exposure	**blur**	**vignetting**	**PRNU**
bias	0.6	-	-	-
$\boldsymbol{\sigma_*}$	0.4	-	-	-

(3) Stereo camera pose

	d	**alpha**	**x**	**y**	**z**	**roll**	**pitch**	**yaw**
unit	m	°	m	m	m	°	°	°
value	10	0	0	0	0	0	0	0
$\boldsymbol{\sigma_{Tremble}}$	-	-	0.05	0.05	0.05	1.0	1.0	1.0

Table 6: Parameters for variation

B.2 - Calibration Parameters

Table 7 lists all applied geometrical calibration parameters. The interior (1) and the stereo-camera calibration (2) are provided by the DLR. To determine the associated uncertainties, the calibration tool [CalLab] was used. Based on the given calibration, multiple poses of the stereo-camera to a chessboard were simulated and applied for a calibration. Different sets of images, each containing 30-40 different poses, have shown slightly different uncertainties. Based on these calibration, realistic achievable uncertainties have been chosen.
The marker calibration (3) was done with a measuring tape. Bulges on the vehicle complicated a precise measurement. The configuration was manually reconstructed in the simulator and the individual marker poses were adjusted to match as best as possible different reference images. Because of this procedure, relatively high uncertainties were assigned. However, the results of Figure 37 prove the quality of this calibration.
The parameter *size* of the marker calibration indicates the diameter of the outer black circle for WhyCon *WC* patterns and the width of the black quad of AprilTag *AT* patterns. The *margins* states the white border applied to each pattern, which is required

(1) Interior camera calibration

	u0	**v0**	**f**	**k1**	**k2**	**k3**	**p1**	**p2**
unit	*px*	*px*	*px*					
CL	703.66	509.4	1960.07	-0.117501	0.225389	0	0	0
CR	725.33	504.38	1963.25	-0.113372	0.197964	0	0	0
$\boldsymbol{\sigma_c}$	0.83	1	1.01	0.004	0.03	0	0	0

(2) Stereo-camera calibration

	x	**y**	**z**	**roll**	**pitch**	**yaw**
unit	*m*	*m*	*m*	°	°	°
$\boldsymbol{H_{CL2CR}}$	-0.3421	0.0019	-0.0003	0.0027	0.0047	-0.0095
$\boldsymbol{\sigma_c}$	0.000126	0.000104	0.000702	0.03337	0.02816	0.00364

(3) Marker calibration

	x	**y**	**z**	**roll**	**pitch**	**yaw**	**width**	**margin**
unit	*m*	*m*	*m*	°	°	°	*m*	*m*
$\boldsymbol{AT_{smallA}}$	-	-	-	-	-	-	0.162	0.02025
$\boldsymbol{AT_{smallB}}$	-	-	-	-	-	-	0.15188	0.02531
$\boldsymbol{AT_{big}}$	-	-	-	-	-	-	0.27494	0.04583
$\boldsymbol{WC_{small}}$	-	-	-	-	-	-	0.1598	0.04
$\boldsymbol{WC_{big}}$	-	-	-	-	-	-	0.2761	0.069
$\boldsymbol{\sigma_m}$	0.01	0.01	0.01	1.0	1.0	1.0	0.001	-

Table 7: Calibration parameters with uncertainties

to ensure the sufficient contrast at the edge of each marker.
The following assignments belong to Figure 21 (p.23).

- Dataset (a) includes markers with dimensions of AT_{smallA}.
- Dataset (b) contains AT_{big}.
- Dataset (c) contains the markers AT_{smallB}, AT_{big}, WC_{small}, WC_{big}.
- Dataset (d) contains only markers of the dimensions of WC_{small}.

The standard deviations σ_{IC}, σ_{CL2CR}, σ_{Marker} represent the uncertainties modeled in the application stage.

B.3 - Dataset Definitions

B.3.1 - Train Dataset

Table 8 defines the measured distances for the real-world dataset of Figure 21 (a,b) that are used as ground truth values. For the measurement a laser scanner [GLM-80] was used. The images of this dataset show large image blooming caused by extrem over exposure. This is exemplary shown in Figure 58.

Dataset (a) - AprilTag Type 1

Desired	**6**	**12**	**21**
Measured	6.0	11.5	21

Dataset (b) - AprilTag Type 2

Desired	**6**	**13**	**24**
Measured	5.88	12.96	24.2

Table 8: Definitions of the real-world dataset of Figure 21 (a,b)

(a) Original (b) 5.88m (c) 12.96m

Figure 58: Comparison of the original AprilTag to exemplary projected markers of the real-world dataset of Figure 21 (a,b)

B.3.2 - Tourneo Dataset

Table 9 defines the measured distances of the real-world experiment of Figure 21 (c) divided in near and far distances. The measurements of dataset (d) are only based on marking on the ground measured with a tape measure. A precise measurement is not required for this particular dataset, because it is only used to estimate the application range.

Dataset (c) - near distance

Desired	**5**	**7.5**	**10**	**12.5**	**15**	**20**
Measured	4.877	7.410	9.952	12.433	14.872	19.934

Dataset (c) - near distance

Desired	**30**	**40**	**50**	**60**	**70**	**80**
Measured	29.711	39.732	49.746	59.759	69.698	80.011

Table 9: Definitions of the real-world dataset of Figure 21 (c)

B.4 - Experiment Completions

B.4.1 - Completion Application Range

alpha = 0°

distance[m]	**10.0**	**20.0**	**30.0**	**40.0**	**50.0**	**60.0**	**70.0**	**80.0**
unit	*%*	*%*	*%*	*%*	*%*	*%*	*%*	*%*
AprilTags+PnP CL	100.0	99.0	56.0	10.0	0.0	0.0	0.0	0.0
WhyCons+PnP CL	100.0	100.0	99.0	96.0	68.0	6.0	0.0	0.0
WhyCon+Circle CL	100.0	100.0	100.0	100.0	100.0	100.0	97.0	86.0
SGM	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

alpha = 30°

distance[m]	**10.0**	**20.0**	**30.0**	**40.0**	**50.0**	**60.0**	**70.0**	**80.0**
unit	*%*	*%*	*%*	*%*	*%*	*%*	*%*	*%*
AprilTags+PnP CL	100.0	100.0	93.0	3.0	0.0	0.0	0.0	0.0
WhyCons+PnP CL	100.0	100.0	100.0	91.0	14.0	0.0	0.0	0.0
WhyCon+Circle CL	100.0	100.0	100.0	100.0	100.0	99.0	89.0	68.0
SGM	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

alpha = 60°

distance[m]	**10.0**	**20.0**	**30.0**	**40.0**	**50.0**	**60.0**	**70.0**	**80.0**
unit	*%*	*%*	*%*	*%*	*%*	*%*	*%*	*%*
AprilTags+PnP CL	100.0	97.0	1.0	0.0	0.0	0.0	0.0	0.0
WhyCons+PnP CL	100.0	93.0	0.0	0.0	0.0	0.0	0.0	0.0
WhyCon+Circle CL	100.0	100.0	100.0	99.0	80.0	29.0	1.0	2.0
SGM	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Table 10: Successrates for experiment of Figure 38 (p. 39)

B.4.2 - Completion Image Exposure

exposure	**0.1**	**0.2**	**0.8**	**1.4**	**1.5**
AprilTags#1+PnP	0.009	-0.001	-0.003	0.07	0.088
AprilTags#3+PnP	0.0	0.0	-0.001	0.006	0.007
WhyCons+PnP	0.001	0.001	0.001	0.001	0.001
WhyCon+Circle	-0.013	-0.01	-0.008	0.045	0.053

Table 11: Bias for experiment of Figure 40 (p.40)

B.4.3 - Completion Uncertainty

Near distances

distance[m]	**5**	**7.5**	**10**	**12.5**	**15**	**20**
unit	*m*	*m*	*m*	*m*	*m*	*m*
AprilTags+PnP CL	0.052	0.071	0.09	0.111	0.131	0.187
WhyCons+PnP CL	0.043	0.061	0.083	0.102	0.121	0.162
WhyCon+Circle CL	0.021	0.03	0.041	0.053	0.064	0.096
SGM	0.057	0.127	0.225	0.353	0.51	0.905

Far distances

distance[m]	**30**	**40**	**50**	**60**	**70**	**80**
unit	*m*	*m*	*m*	*m*	*m*	*m*
AprilTags+PnP CL	-	-	-	-	-	-
WhyCons+PnP CL	0.238	0.318	-	-	-	-
WhyCon+Circle CL	0.205	0.43	0.707	0.692	1.41	3.008
SGM	-	3.803	6.04	8.598	12.182	16.811

Table 12: Standard uncertainties for experiment of Figure 42 (p.42)

B.4.4 - Completion Correlation

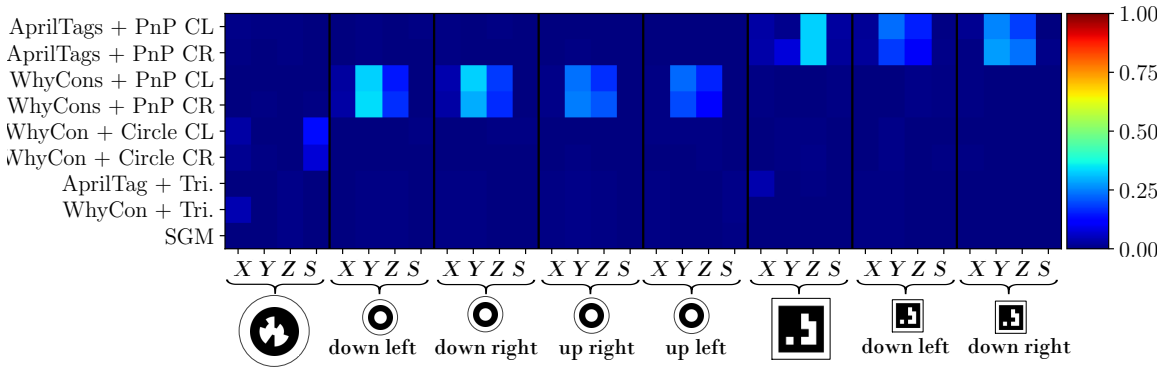


Figure 59: Correlation to uncertainty of the marker pose (Figure 44, p.44)

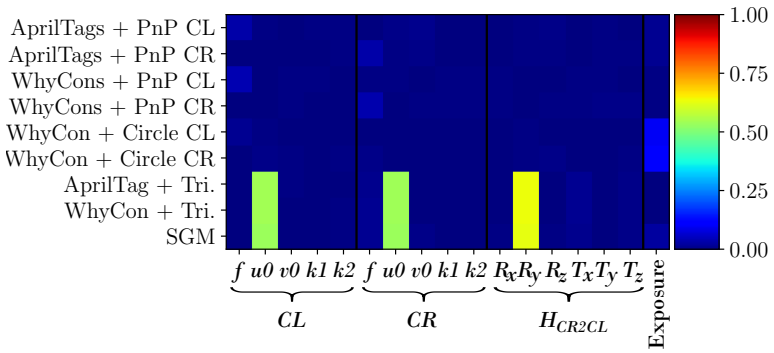


Figure 60: Correlation to camera calibration uncertainty (Figure 45, p.44)

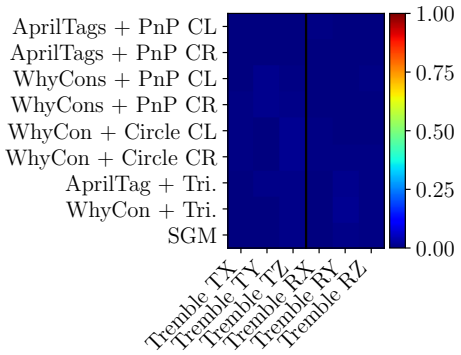


Figure 61: Correlation to orientation trembling. Please note that the trembling is part of the ground truth matrix.

Corr.	**WC-d-l**	**WC-d-r**	**WC-t-r**	**WC-t-l**	**AT-t**	**AT-d-l**	**AT-d-r**
X[m]	0.03	0.04	0.009	0.008	0.035	0.023	0.018
Y[m]	-0.333	0.33	0.239	-0.23	-0.013	-0.23	0.256
Z[m]	0.147	0.183	-0.171	-0.16	-0.332	0.154	0.186
Angle[◦]	-23.8	29.1	-35.6	34.9	87.88	-33.57	35.97

Table 13: The columns of "WC" correspond to the correlation of the WhyCons+PnP and the four whycon marker. The columns of "AT" correspond to the correlation of the AprilTag+PnP and the three AprilTags. The angle is defined in the yz-plane. (m-middle, d- down, t-top, l-left, r-right)

B.4.5 - Completion Marker Uncertainty

s_m	**0.0**	**1.0**	**2.0**	**3.0**
unit	m	m	m	m
AprilTags+PnP CL	0.005	0.088	0.175	0.265
WhyCons+PnP CL	0.005	0.08	0.162	0.24
WhyCon+Circle CL	0.018	0.041	0.076	0.111
SGM	0.226	0.227	0.229	0.222

Table 14: Standard deviations for experiment of Figure 47 (p. 45)

B.4.6 - Completion Camera Uncertainty

s_c	**0.0**	**0.333**	**0.667**	**1.0**
unit	m	m	m	m
AprilTags+PnP CL	0.088	0.087	0.089	0.088
WhyCons+PnP CL	0.082	0.08	0.082	0.082
WhyCon+Circle CL	0.041	0.04	0.041	0.041
SGM	0.017	0.078	0.149	0.229

Table 15: Standard deviations for experiment of Figure 48 (p. 46)

B.4.7 - Completion Influence of the Baseline

baseline[m]	**0.34**	**0.68**	**1.02**	**1.36**
unit	m	m	m	m
AprilTags+PnP CL	0.185	0.192	0.191	0.19
WhyCons+PnP CL	0.164	0.159	0.162	0.162
WhyCon+Circle CL	0.099	0.096	0.097	0.098
SGM	0.913	0.454	0.302	-

Table 16: Standard deviations for experiment of Figure 49 (p. 46)

B.4.8 - Completion Aggregation

Near distances

distance[m]	**30**	**40**	**50**	**60**	**70**	**80**
unit	*m*	*m*	*m*	*m*	*m*	*m*
WhyCon+Circle CL	0.205	0.43	0.707	0.692	1.41	3.008
WhyCon+Circle CR	0.209	0.422	0.695	0.709	1.436	3.069
WhyCon+Tri.	2.056	3.662	5.803	8.445	11.95	16.204
SGM	2.078	3.817	6.0	8.654	12.236	16.771
Combined	0.18	0.349	0.543	0.573	1.103	3.021

Far distances

distance[m]	**5**	**7.5**	**10**	**12.5**	**15**	**20**
unit	*m*	*m*	*m*	*m*	*m*	*m*
WhyCon+Circle CL	0.021	0.03	0.041	0.053	0.064	0.096
WhyCon+Circle CR	0.021	0.03	0.04	0.053	0.064	0.096
WhyCon+Tri.	0.057	0.126	0.223	0.347	0.504	0.897
SGM	-	0.127	0.225	0.353	0.51	0.905
Combined	-	0.029	0.039	0.051	0.062	0.091

Table 17: Standard uncertainties for experiment of Figure 51 (p. 48)